



# **Intel® SGX Software Installation Guide**

## **For Linux\* OS**

Revision <1.1>

<9/18/2024>

# Table of Contents

---

Table of Contents .....	2
Introduction .....	3
Installation Instructions - QuickStart .....	4
Driver Installation .....	4
User Mode Software Installation .....	9
Intel® SGX Application User.....	9
Intel® SGX Application Developer .....	18
Building the Intel® SGX Software Stack.....	22
Intel® Software Guard Extensions – Software Packages .....	23
Access to AESM Plug-ins.....	25
Detailed Description of Packages.....	26
FIPS 140-3 Certifiable mode .....	28
Summary of Use Cases and Required Packages.....	30
Intel® SGX Software Development Kit for Linux* OS .....	33
Appendix 1: Advanced Configuration Topics .....	35
Access to ECDSA Quote Generation using the DCAP Quoting Library .....	35
Configuration of AESM Proxy Service.....	38
Start AESM without systemd and syslog, and other command line options .....	38
Appendix 2: How to setup Provisioning Certificate Caching Service (PCCS) on local machine .....	39
Disclaimer and Legal Information .....	41

# Introduction

---

This document describes installation of the Intel® Software Guard Extensions (Intel® SGX) Software Components, specifically:

- Intel® SGX Software Development Kit (SDK), which aids Software Developers in creating applications that use Intel SGX Technology.
- Intel® SGX Platform Software (PSW) for Linux\* OS, which provides software modules to run Intel® SGX applications on the Linux\* OS.
- Intel® SGX Data Center Attestation Primitives (DCAP) for Linux\* OS, which provides software modules to aid Intel® Applications in performing attestation within the data center.

Installation packages are provided as binary installers for the SDK and PSW at <https://download.01.org/intel-sgx/latest/linux-latest/distro/>. In addition, repositories are supported to distribute packages for the following OSs:

- Ubuntu and Debian: PSW and DCAP packages are provided in a Debian\* repository at [https://download.01.org/intel-sgx/sgx\\_repo/<distro>/](https://download.01.org/intel-sgx/sgx_repo/<distro>/) and via a tar file located at <https://download.01.org/intel-sgx/latest/linux-latest/distro/<distro>/>.
- Red Hat, CentOS, Anolis, and SUSE: PSW and DCAP packages are provided as RPM Packages via a tar file located in the corresponding folder in <https://download.01.org/intel-sgx/latest/linux-latest/distro/<distro>/>.

This document focuses on describing the installation process for the below Linux distributions:

- Red Hat\* Enterprise Linux\*
- CentOS\* Server
- Ubuntu\* Server
- SUSE Linux Enterprise Server
- Anolis OS
- Debian

See the Release Notes provided at <https://01.org/intel-software-guard-extensions/downloads> for specific information about each Linux distribution support.

The source code for Intel® SGX Software Components is provided in two locations on GitHub\*:

- Intel SGX SDK and PSW are provided at: <https://github.com/intel/linux-sgx>
- Intel DCAP is provided at: <https://github.com/intel/SGXDataCenterAttestationPrimitives>

## Installation Instructions - QuickStart

---

The installation of the Intel® Software Guard Extensions (Intel® SGX) software packages for Ubuntu\* OS or Red Hat\* Enterprise Linux\* begins with the installation of the proper Intel® SGX Driver. After you install the driver, you can install the software packages.

This section is designed to provide quick setup instructions to help with configuring a platform to support SGX for a variety of usages – running an Intel® SGX application, building and running an Intel® SGX application, or building the Intel® SGX software stack. For detailed explanation of packages, see Intel® Software Guard Extensions – Software Packages.

### Driver Installation

A key component of a platform running Intel® SGX Enclaves is a kernel mode driver, which is instrumental in the loading and managing an Intel® SGX enclave. To understand how to configure a platform, you must be aware of Launch Control provided in Intel® SGX platforms. Launch Control refers to the methods and restrictions that the platform puts on controlling, which may provide tokens to applications for launching Intel® SGX enclaves. There are two characteristic configurations for Launch Control:

- *Flexible Launch Control:* platforms, including virtual machines, that support Intel® SGX Launch Control may be configured for a specific enclave signer by the kernel mode driver. In many cases, the driver dynamically reconfigures launch control for each enclave loaded, so that the enclave does not need a valid Launch Token to run (see EINITOKEN in the [Intel® 64 and IA-32 Architectures Software Developer Manuals](#) for more information on Launch Token).
- *Legacy Launch Control:* platforms that do not support Intel® SGX Launch Control or that are not configured by the BIOS for Flexible Launch Control may still load enclaves using Legacy Launch Control. In this mode, the platform must use an Intel-signed Launch Enclave to provide a Launch Token to the driver for loading the enclave. Also, the enclave owner's MRSIGNER (a hash of the owner's public signing key) must be added to Intel's Launch Policy List to run an enclave in release mode (For more information, see <https://software.intel.com/en-us/sgx/request-license>). Note: Though Legacy Launch Control restricts what enclaves can run in release mode, it allows an enclave to run in SGX debug mode regardless of whether the enclave owner's MRSIGNER has been added to the Launch Policy (see [Intel® 64 and IA-32 Architectures Software Developer Manuals](#) for information on SGX debug mode enclaves). This allows to develop and debug test-signed debug enclaves.

There are currently three different drivers that can be used to support Intel® SGX. The platform must be configured with only one of these drivers:

1. *In-kernel Driver* (/dev/{sgx\_enclave, sgx\_provision}): Mainline kernel release 5.11 or higher includes the SGX In-Kernel driver. The In-Kernel Driver requires the platform to support and to be configured for Flexible Launch Control.
2. *DCAP Driver* (/dev/{sgx\_enclave, sgx\_provision}): The goal of the DCAP driver is to provide an interface close to the In-kernel Driver in order to provide Intel® SGX support to Linux OSs that do not have the Intel® SGX driver built into the kernel. This driver also requires the platform to support and to be configured for Flexible Launch Control. Since this driver supports an out-of-tree implementation, it does not include new features supported by later kernel releases such as EDMM, SGX KVM, SGX cgroups, and NUMA aware EPC allocation. This driver is only updated for necessary security fixes. The in-kernel driver should be used whenever it is possible.
3. *Out-of-tree Driver* (/dev/isgx): This driver is provided to support running Intel® SGX enclaves on platforms that only support Legacy Launch Control. It may also be installed on platforms configured with Flexible Launch Control; however, then these platforms will only load enclaves that conform to the Legacy Launch Control Policy.

The process of selection and installation of each driver is covered in the following sections.

## In-Kernel Driver Module Installation

The In-kernel driver is provided with Linux\* kernel beginning with the version 5.11. Developers and platform owners can install the 5.11 (or higher) kernel binaries provided by their Linux distribution or build the kernel from source code following the steps described in <https://wiki.ubuntu.com/KernelTeam/GitKernelBuild>. SGX option CONFIG\_X86\_SGX must be enabled in the `make menuconfig` step, otherwise the SGX module cannot be built. Other SGX configuration options such as CONFIG\_X86\_SGX\_KVM might be enabled depending on the user needs. Earlier versions of the kernel can be adapted with a patch. The latest Intel-SGX Patch to the kernel can be obtained at <https://patchwork.kernel.org/project/intel-sgx/list/>. This document does not cover the process of patching and building a kernel.

## DCAP Driver Installation

The DCAP Driver is the recommended driver to use on the Linux kernel version between 4.15 and 5.6 inclusive and on platforms that support and are configured for Flexible Launch Control. Installing DCAP driver on kernel 5.11 or higher with SGX In-Kernel driver gives the build error message, “Can't install DCAP SGX driver with in-kernel SGX support”.

### *Prerequisites*

Since the DCAP Driver is built from the driver package, install the required components that support the Intel® SGX PSW installation. The following procedure is used to install the required components on each Linux distribution.

**Note:** This following command lines contain modules that are needed in components described in subsequent sections of this document

## Ubuntu and Debian

```
sudo apt-get install build-essential ocaml automake autoconf libtool wget  
python3 libssl-dev dkms
```

```
sudo update-alternatives --install /usr/bin/python python /usr/bin/python3  
1
```

## Red Hat Enterprise Linux

```
sudo subscription-manager repos --enable codeready-builder-for-rhel-8-  
x86_64-rpms
```

```
sudo yum install https://dl.fedoraproject.org/pub/epel/epel-release-  
latest-8.noarch.rpm
```

```
sudo yum install openssl-devel libcurl-devel yum-utils wget dkms make  
kernel-devel kernel
```

## CentOS Stream 9

```
sudo dnf group install 'Development Tools'
```

```
sudo dnf install ocaml ocaml-ocamlbuild redhat-rpm-config openssl-devel  
wget rpm-build git cmake python3 dkms
```

CentOS 8.3 sudo dnf group install 'Development Tools'

```
sudo dnf --enablerepo=powertools install ocaml ocaml-ocamlbuild openssl-  
devel wget cmake python3 dkms
```

```
sudo alternatives --set python /usr/bin/python3
```

## Anolis

```
sudo dnf group install 'Development Tools'
```

```
sudo dnf --enablerepo=PowerTools install ocaml ocaml-ocamlbuild openssl-  
devel wget cmake python3 dkms
```

```
sudo alternatives --set python /usr/bin/python3
```

## SUSE Linux Enterprise Server

```
sudo zypper install --type pattern devel_basis
```

```
sudo zypper install ocaml ocaml-ocamlbuild automake autoconf libtool wget  
python3 libopenssl-devel cmake dkms
```

```
sudo update-alternatives --install /usr/bin/python python /usr/bin/python3  
1
```

### ***Install DCAP Driver***

1. Download the latest Intel SGX Driver binary file the appropriate distro directory:

<https://download.01.org/intel-sgx/latest/linux-latest/distro>

```
wget https://download.01.org/intel-sgx/latest/dcap-latest/linux/distro/  
${distro_name}/sgx_linux_x64_driver_${version}.bin
```

2. Set the protections to allow for the .bin file execution:

```
chmod 777 sgx_linux_x64_driver_${version}.bin
```

3. Install the driver:

```
sudo ./sgx_linux_x64_driver_${version}.bin
```

The installer also loads the DCAP Driver and sets it to auto-load when the system reboots.

To verify the driver loaded correctly, check that the device shows up in the /dev folder:

```
ls -la /dev/sgx*  
crw-----. 1 root root 10, 60 Mar 18 15:06 /dev/sgx_enclave  
crw-----. 1 root root 10, 59 Mar 18 15:06 /dev/sgx_provision
```

After the DCAP Driver installation, you can see a generated script `uninstall.sh` in the `/opt/intel/sgxdriver` directory. Use this script to uninstall the driver.

## **Out-of-Tree Driver Installation**

The Out-of-Tree Driver is recommended for use on platforms that do not support or are not configured for Flexible Launch Control. It is not recommended to install the legacy Out-of-Tree driver on FLC-enabled platforms. If the legacy Out-of-Tree driver is installed on an FLC platform running with Kernel 5.11 or higher, the installation succeeds; however, Intel SGX PSW utilizes only the In-Kernel Driver.

### ***Prerequisites***

Since the Out-of-Tree Driver is built from the driver package, install the required components that support the Intel® SGX PSW installation.

#### Ubuntu and Debian

On Debian:

```
sudo apt-get install build-essential ocaml automake autoconf libtool wget  
python3 libssl-dev  
  
sudo update-alternatives --install /usr/bin/python python /usr/bin/python3  
1
```

On Ubuntu 20.04, Ubuntu 22.04 and Ubuntu 24.04:

```
sudo apt-get install build-essential ocaml automake autoconf libtool wget  
python-is-python3 libssl-dev
```

### Red Hat Enterprise Linux

```
sudo subscription-manager repos --enable codeready-builder-for-rhel-8-  
x86_64-rpms
```

```
sudo yum install https://dl.fedoraproject.org/pub/epel/epel-release-  
latest-8.noarch.rpm
```

```
sudo yum install openssl-devel libcurl-devel yum-utils wget dkms make  
kernel-devel kernel
```

### CentOS Stream 9

```
sudo dnf group install 'Development Tools'
```

```
sudo dnf install ocaml ocaml-ocamlbuild redhat-rpm-config openssl-devel  
wget rpm-build git cmake python3
```

### CentOS 8.3

```
sudo dnf group install 'Development Tools'
```

```
sudo dnf --enablerepo=powertools install ocaml ocaml-ocamlbuild openssl-  
devel wget cmake python3
```

```
sudo alternatives --set python /usr/bin/python3
```

### Anolis

```
sudo dnf group install 'Development Tools'
```

```
sudo dnf --enablerepo=PowerTools install ocaml ocaml-ocamlbuild openssl-  
devel wget cmake python3
```

```
sudo alternatives --set python /usr/bin/python3
```

### SUSE Linux Enterprise Server

```
sudo zypper install --type pattern devel_basis
```

```
sudo zypper install ocaml ocaml-ocamlbuild automake autoconf libtool wget  
python3 libopenssl-devel cmake
```

```
sudo update-alternatives --install /usr/bin/python python /usr/bin/python3  
1
```



### ***Install Out-of-Tree Driver***

1. Download the latest Intel® SGX Driver binary file from the distro directory:

<https://download.01.org/intel-sgx/latest/linux-latest/distro/>

```
wget https://download.01.org/intel-sgx/latest/linux-latest/distro/${distro_name}/sgx_linux_x64_driver_${version}.bin
```

2. Set protections to allow for the .bin file execution:

```
chmod 777 sgx_linux_x64_driver_${version}.bin
```

3. Install the driver:

```
sudo ./sgx_linux_x64_driver_${version}.bin
```

The installer also loads the Out-of-Tree Driver and sets it to be `auto-load` when the system reboots.

To verify the driver loaded correctly, check that the device shows up in the `/dev` folder:

```
ls -la /dev/isgx
```

```
crw-rw-rw- 1 root root 241, 0 Mar 10 10:45 /dev/isgx
```

After the Out-of-Tree Driver installation, you can see a generated script `uninstall.sh` in the `/opt/intel/sgxdriver` directory. Use this script to uninstall the driver.

## **User Mode Software Installation**

The procedure for configuring a platform with the necessary Intel® SGX software packages, provided as binary installers and/or Debian packages, depends on the intended use of the platform. Choose the role that describes your needs best:

- **Intel® SGX Application User:** install an Intel® SGX application that runs an Intel® SGX enclave on the system.
- **Intel® SGX Application Developer:** build or develop an Intel® SGX application that runs an Intel® SGX enclave on the system.
- **Intel® SGX Software Stack Developer or Builder:** build or develop the Intel® SGX Software Stack: The Intel® SGX Software Development Kit (Intel® SGX SDK), the Intel® SGX Platform Software (Intel® SGX PSW), or the Intel® SGX Data Center Attestation Primitives (Intel® SGX DCAP).

This section provides shortcuts on system configuration for the needs described above.

### **Intel® SGX Application User**

To run an Intel SGX Application built with the Intel® SGX SDK, install appropriate packages from the Intel® SGX Platform Software (Intel® SGX PSW) and Intel® SGX DCAP. They are provided as Debian\* packages.

#### ***Ubuntu***

To configure the system to run an Intel® SGX application:

1. Install the following Debian Library and App Packages: `libsgx-epid`, `libsgx-quote-ex`, `libsgx-dcap-ql`, which also installs the dependent packages `libsgx-urts`, `libsgx-launch`, `libsgx-ae-le`, `libsgx-ae-pce`, `libsgx-ae-qe3`, `libsgx-ae-qve`, `libsgx-ae-epid`, `libsgx-qe3-logic`, `libsgx-pce-logic`, `libsgx-dcap_quote-verify`, `libsgx-aesm-ecdsa-plugin`, `libsgx-aesm-epid-plugin`, `libsgx-ae-launch-plugin`, `libsgx-aesm-quote-ex-plugin`, `libsgx-enclave-common`, and `sgx-aesm-service` following these steps:

- a. Ensure you have connection to the internet and open a terminal.

- b. Add the following repository to your sources:

- i. For Ubuntu\* 20.04:

```
echo 'deb [arch=amd64] https://download.01.org/intel-sgx/sgx_repo/ubuntu focal main' | sudo tee /etc/apt/sources.list.d/intel-sgx.list
```

- ii. For Ubuntu\* 22.04:

```
echo 'deb [signed-by=/etc/apt/keyrings/intel-sgx-keyring.asc arch=amd64] https://download.01.org/intel-sgx/sgx_repo/ubuntu jammy main' | sudo tee /etc/apt/sources.list.d/intel-sgx.list
```

- iii. For Ubuntu\* 24.04:

```
echo 'deb [signed-by=/etc/apt/keyrings/intel-sgx-keyring.asc arch=amd64] https://download.01.org/intel-sgx/sgx_repo/ubuntu noble main' | sudo tee /etc/apt/sources.list.d/intel-sgx.list
```

- c. **(Optional)** If you have previously configured the system with an old Debian repository public key for SGX project that has expired, you will need to remove the old public key before proceeding to the next step. To do so, run the following commands:

- i. For Ubuntu\* 20.04:

```
sudo apt-key del <keyid>
```

The <keyid> is the last 8 characters of the expired key's fingerprint. The expired key could be found with the command `apt-key list | grep "expired" `.`

- ii. For Ubuntu\* 22.04 and Ubuntu\* 24.04:

```
sudo rm -f /etc/apt/keyrings/intel-sgx-keyring.asc
```

- d. Get the Debian repo public key and add it to the list of trusted keys that are used by `apt` to authenticate packages:

- i. For Ubuntu\* 20.04:

```
wget -qO - https://download.01.org/intel-sgx/sgx_repo/ubuntu/intel-sgx-deb.key | sudo apt-key add
```

- ii. For Ubuntu\* 22.04 and Ubuntu\* 24.04:

```
wget https://download.01.org/intel-sgx/sgx_repo/ubuntu/intel-sgx-deb.key
cat intel-sgx-deb.key | sudo tee /etc/apt/keyrings/intel-sgx-keyring.asc > /dev/null
```

- e. Update apt and install the following packages:

```
sudo apt-get update
sudo apt-get install libsgx-epid libsgx-quote-ex libsgx-dcap-ql
```

- f. **(Optional)** To debug with `sgx-gdb`, install the debug symbol package. For Ubuntu, the debug symbols are included in the following packages:

```
sudo apt-get install libsgx-urts-dbgsym libsgx-enclave-common-dbgsym libsgx-dcap-ql-dbgsym libsgx-dcap-default-ql-dbgsym
```

- g. **(Alternate Installation Method)** Another method is to set up a local or on disk Ubuntu repo using the tarball that contains all the packages. You should download the tar file that matches your Ubuntu version.

On a platform with Internet access, download the tar file:

```
wget -qO - https://download.01.org/intel-sgx/latest/linux-latest/distro/<distro>/sgx_debian_local_repo.tgz
```

Once those two files are available on the target platform:

```
tar xzf sgx_debian_local_repo.tgz
```

- i. For Ubuntu\* 20.04:

```
echo 'deb [trusted=yes arch=amd64]
file:///path/to/ubuntu_debian_repo focal main' | sudo tee
/etc/apt/sources.list.d/sgx-repo.list
```

- ii. For Ubuntu\* 22.04:

```
echo 'deb [trusted=yes arch=amd64]
file:///path/to/ubuntu_debian_repo jammy main' | sudo tee
/etc/apt/sources.list.d/sgx-repo.list
```

- iii. For Ubuntu\* 24.04:

```
echo 'deb [trusted=yes arch=amd64]
file:///path/to/ubuntu_debian_repo noble main' | sudo tee
/etc/apt/sources.list.d/sgx-repo.list
```

```
sudo apt-get update
```

Then install the SGX packages you need using the `apt-get` command. For example:

```
sudo apt-get install -y libsgx-urts libsgx-launch libsgx-enclave-common
```

- h. **(Optional)** If you intend to run an SGX application that loads an enclave requiring the Provision Key Access, the user needs to be added to the group "sgx\_prv". Applications that obtain a quote from the DCAP Quote Generation library for the purposes of remote attestation may require Provision Key Access. Use the following command to add access for a user:

```
sudo usermod -aG sgx_prv <username>
```

- i. **(Optional)** To run enclave apps on distros with default "sgx" group, the user needs to be added to the group of "sgx ". Use the following command to add access for a user:

```
sudo usermod -aG sgx <username>
```

- 2. Set up the Intel® Software Guard Extensions Data Center Attestation Primitives (Intel® SGX DCAP), Provisioning Certificate Caching Service (PCCS) and Quote Provider Library (QPL). The PCCS and QPL work together to first cache DCAP attestation collateral information and then make the information available to the DCAP Quote Generation Library (libsgx-dcap-ql). These packages are provided as reference designs that users may employ. If you are running in a specific cloud, check with the CSP to see if it provides a CSP-specific QPL that accesses a custom provisioning certificate caching service in that CSP's cloud.

- i. **[Optional]** Setup the Provisioning Certificate Caching Service (PCCS):  
If the CSP already provides the caching service, you don't need to setup it by yourself. Please consult Cloud Service Provider about how to use their caching service. If you are not running in the cloud or there is no caching service available, please refer to [Appendix 2](#) for how to setup local caching service.

- ii. Install the DCAP QPL package (If the CSP has a specific QPL then you need to install that one):

```
sudo apt-get install libsgx-dcap-default-qpl
```

- 3. **Note on upgrading from a legacy installation:** Before release version 2.8, the Intel® SGX PSW was installed from a single package named `libsgx-enclave-common`. Starting with the 2.8 release, the Intel® SGX PSW is split into many smaller packages with individual libraries. `libsgx-enclave-common` now only contains the `libsgx-enclave-common` library. As a result, a simple upgrade will end up with a subset of the Intel® SGX PSW being installed on the system. You need to install the additional packages listed above. In addition, you will encounter some error message when you attempt to install the extra packages with an older version of `libsgx-enclave-common`, because the older `libsgx-enclave-common` package have files attributed to it that are now installed from the new packages. The suggested method to upgrade is:

- i. Uninstall old versions of sgx packages:

```
sudo apt-get remove '^sgx-.*' '^libsgx-.*'
```

- ii. Update and install the appropriate packages:

```
sudo apt update
```

```
sudo apt install <package_name>
```

## Debian

To configure the system to run an Intel® SGX application:

1. Install the following Debian Library and App Packages: `libsgx-epid`, `libsgx-quote-ex`, `libsgx-dcap-ql`, which also installs the dependent packages `libsgx-urts`, `libsgx-launch`, `libsgx-ae-le`, `libsgx-ae-pce`, `libsgx-ae-qe3`, `libsgx-ae-qve`, `libsgx-ae-epid`, `libsgx-qe3-logic`, `libsgx-pce-logic`, `libsgx-dcap_quote-verify`, `libsgx-aesm-ecdsa-plugin`, `libsgx-aesm-epid-plugin`, `libsgx-ae-launch-plugin`, `libsgx-aesm-quote-ex-plugin`, `libsgx-enclave-common`, and `sgx-aesm-service` following these steps:

- a. Find Debian packages for SGX libraries and services, which are currently provided in a single TAR archive at

<https://download.01.org/intel-sgx/latest/linux-latest/distro/<distro>/>

- b. Download the file `sgx_debian_local_repo.tgz` to a selected folder, for example `/opt/intel`

```
cd /opt/intel
sudo wget https://download.01.org/intel-sgx/latest/linux-latest/distro/<distro>/sgx_debian_local_repo.tgz
```

- c. Verify the downloaded repo file with the SHA value in this file:

[https://download.01.org/intel-sgx/latest/dcap-latest/linux/SHA256SUM\\_dcap\\_<version>.cfg](https://download.01.org/intel-sgx/latest/dcap-latest/linux/SHA256SUM_dcap_<version>.cfg)

```
sha256sum sgx_debian_local_repo.tgz
```

- d. Expand the archive:

```
sudo tar xvf sgx_debian_local_repo.tgz
```

- e. Add the repository to your sources:

```
echo 'deb [trusted=yes arch=amd64]
/opt/intel/sgx_debian_local_repo buster main' | sudo tee
/etc/apt/sources.list.d/sgx_debian_local_repo.list
```

- f. Update apt and install the following packages:

```
sudo apt-get update
sudo apt-get install libsgx-epid libsgx-quote-ex libsgx-dcap-ql
```

- g. **(Optional)** To debug with `sgx-gdb`, install the debug symbol package.

```
sudo apt-get install libsgx-urts-dbgsym libsgx-enclave-common-
dbgsym libsgx-dcap-ql-dbgsym libsgx-dcap-default-ql-dbgsym
```

- h. **(Optional)** If you intend to run an SGX application that loads an enclave requiring the Provision Key Access, the user needs to be added to the group `"sgx_prv"`. Applications

that obtain a quote from the DCAP Quote Generation library for the purposes of remote attestation may require Provision Key Access. Use the following command to add access for a user:

```
sudo usermod -aG sgx_prv <username>
```

- i. **(Optional)** To run enclave apps on distros with default "sgx" group, the user needs to be added to the group of "sgx ". Use the following command to add access for a user:

```
sudo usermod -aG sgx <username>
```

2. Set up the Intel® Software Guard Extensions Data Center Attestation Primitives (Intel® SGX DCAP), Provisioning Certificate Caching Service (PCCS) and Quote Provider Library (QPL). The PCCS and QPL work together to first cache DCAP attestation collateral information and then make the information available to the DCAP Quote Generation Library (libsgx-dcap-ql). These packages are provided as reference designs that users may employ. If you are running in a specific cloud, check with the CSP to see if it provides a CSP-specific QPL that accesses a custom provisioning certificate caching service in that CSP's cloud.

- i. **[Optional]** Setup the Provisioning Certificate Caching Service (PCCS):  
If the CSP already provides the caching service, you don't need to setup it by yourself. Please consult Cloud Service Provider about how to use their caching service. If you are not running in the cloud or there is no caching service available, please refer to [Appendix 2](#) for how to setup local caching service.
- ii. Install the DCAP QPL package (If the CSP has a specific QPL then you need to install that one):

```
sudo apt-get install libsgx-dcap-default-qpl
```

3. **Note on upgrading from a legacy installation:** Before release version 2.8, the Intel® SGX PSW was installed from a single package named `libsgx-enclave-common`. Starting with the 2.8 release, the Intel® SGX PSW is split into many smaller packages with individual libraries. `libsgx-enclave-common` now only contains the `libsgx-enclave-common` library. As a result, a simple upgrade will end up with a subset of the Intel® SGX PSW being installed on the system. You need to install the additional packages listed above. In addition, you will encounter some error message when you attempt to install the extra packages with an older version of `libsgx-enclave-common`, because the older `libsgx-enclave-common` package have files attributed to it that are now installed from the new packages. The suggested method to upgrade is:

- i. Uninstall old versions of sgx packages:

```
sudo apt-get remove '^sgx-.*' '^libsgx-.*'
```

- ii. Update and install the appropriate packages:

```
sudo apt update
```

```
sudo apt install <package_name>
```

### **Red Hat, CentOS and Anolis**

To configure the system to run an Intel® SGX application:

1. Install the following RPM Packages for `libsgx-urts`, `libsgx-launch`, `libsgx-epid`, `libsgx-quote-ex`, `libsgx-dcap-ql`; that also install the dependent packages `libsgx-ae-le`, `libsgx-ae-pce`, `libsgx-ae-qe3`, `libsgx-ae-qve`, `libsgx-ae-epid`, `libsgx-qe3-logic`, `libsgx-pce-logic`, `libsgx-dcap_quote-verify`, `libsgx-aesm-ecdsa-plugin`, `libsgx-aesm-epid-plugin`, `libsgx-ae-launch-plugin`, `libsgx-aesm-quote-ex-plugin`, `libsgx-enclave-common`, and `sgx-aesm-service` with the following method:

- a. Find RPM packages for SGX libraries and services, which are currently provided in a single TAR archive at

```
https://download.01.org/intel-sgx/latest/linux-latest/distro/<distro>/
```

- b. Download the file `sgx_rpm_local_repo.tgz` to a selected folder, for example `/opt/intel`

```
cd /opt/intel
sudo wget https://download.01.org/intel-sgx/latest/linux-latest/distro/<distro>/sgx\_rpm\_local\_repo.tgz
```

- c. Verify the downloaded repo file with the SHA value in this file:  
[https://download.01.org/intel-sgx/latest/dcap-latest/linux/SHA256SUM\\_dcap\\_<version>.cfg](https://download.01.org/intel-sgx/latest/dcap-latest/linux/SHA256SUM_dcap_<version>.cfg)

```
sha256sum sgx_rpm_local_repo.tgz
```

- d. Expand the archive:

```
sudo tar xvf sgx_rpm_local_repo.tgz
```

- e. Add the RPM local repository to your local repository list

```
sudo yum-config-manager --add-repo file://PATH_TO_LOCAL_REPO
```

- f. Install all the latest packages using `'sudo dnf --nogpgcheck1 install <package names>'`. Here is an example that installs all packages from the 2.11 release with DCAP 1.8.

```
sudo dnf --nogpgcheck install libsgx-urts libsgx-launch libsgx-epid libsgx-quote-ex libsgx-dcap-ql
```

- g. **(Optional)** To debug with `sgx-gdb`, install `*-debuginfo` packages to get the debug symbol. For example:

```
sudo dnf --nogpgcheck install --repofrompath=SGX, /opt/intel/sgx_rpm_local_repo libsgx-aesm-ecdsa-plugin-debuginfo libsgx-aesm-epid-plugin-debuginfo libsgx-aesm-launch-plugin-debuginfo libsgx-aesm-pce-plugin-debuginfo libsgx-aesm-quote-ex-plugin-debuginfo libsgx-dcap-default-ql-debuginfo libsgx-dcap-ql-debuginfo libsgx-dcap-quote-verify-debuginfo libsgx-enclave-common-debuginfo libsgx-
```

---

<sup>1</sup> Since the local repo is not signed with GPG, ignore the GPG check when installing the packages.

```
epid-debuginfo libsgx-launch-debuginfo libsgx-pce-logic-  
debuginfo libsgx-qe3-logic-debuginfo libsgx-quote-ex-  
debuginfo libsgx-ra-network-debuginfo libsgx-ra-uefi-  
debuginfo libsgx-urts-debuginfo sgx-aesm-service-debuginfo  
sgx-pck-id-retrieval-tool-debuginfo sgx-ra-service-  
debuginfo
```

- h. **(Optional)** To run an Intel® SGX application that loads an enclave requiring the Provision Key Access, the user needs to be added to the group of "sgx\_prv". This is true for any application that obtains a quote from the DCAP Quote Generation library for the purposes of remote attestation. Use the following command to add access for a user:

```
sudo usermod -aG sgx_prv <username>
```

- i. **(Optional)** To run enclave apps on distros with default "sgx" group, the user needs to be added to the group of "sgx ". Use the following command to add access for a user:

```
sudo usermod -aG sgx <username>
```

2. Set up the Intel® Software Guard Extensions Data Center Attestation Primitives (Intel® SGX DCAP) Provisioning Certificate Caching Service (PCCS) and Quote Provider Library (QPL). The PCCS and QPL work together to first cache DCAP attestation collateral information and then make the information available to the DCAP Quote Generation Library (libsgx-dcap-ql). These packages are provided as reference designs that users may employ. If you are running in a specific cloud, check with the CSP to see if it provides a CSP-specific QPL that accesses a custom provisioning certificate caching service in that CSP's cloud.
  - a. [Optional] Setup the Provisioning Certificate Caching Service (PCCS):  
If the CSP already provides the caching service, you don't need to setup it by yourself. Please consult Cloud Service Provider about how to use their caching service. If you are not running in the cloud or there is no caching service available, please refer to [Appendix 2](#) for how to setup local caching service.
  - b. Install the DCAP QPL package (If the CSP has a specific QPL then you need to install that one):  

```
sudo yum install --nogpgcheck libsgx-dcap-default-ql
```

### ***SUSE Linux Enterprise Server***

To configure the system to run an Intel® SGX application:

1. Install the following RPM Packages for libsgx-urts, libsgx-launch, libsgx-epid, libsgx-quote-ex, libsgx-dcap-ql; that also install the dependent packages libsgx-ae-le, libsgx-ae-pce, libsgx-ae-qe3, libsgx-ae-qve, libsgx-ae-epid, libsgx-qe3-logic, libsgx-pce-logic, libsgx-dcap\_quote-verify, libsgx-aesm-ecdsa-plugin, libsgx-aesm-epid-plugin, libsgx-ae-launch-plugin, libsgx-aesm-quote-ex-plugin, libsgx-enclave-common, and sgx-aesm-service with the following method:



- a. Find RPM packages for SGX libraries and services, which are currently provided in a single TAR archive at

```
https://download.01.org/intel-sgx/latest/linux-latest/distro/<distro>/
```

- b. Download the file `sgx_rpm_local_repo.tgz` to a selected folder, for example `/opt/intel`

```
cd /opt/intel
sudo wget https://download.01.org/intel-sgx/latest/linux-latest/distro/<distro>/sgx_rpm_local_repo.tgz
```

- c. Verify the downloaded repo file with the SHA value in this file:  
[https://download.01.org/intel-sgx/latest/dcap-latest/linux/SHA256SUM\\_dcap\\_<version>.cfg](https://download.01.org/intel-sgx/latest/dcap-latest/linux/SHA256SUM_dcap_<version>.cfg)

```
sha256sum sgx_rpm_local_repo.tgz
```

- d. Expand the archive:

```
sudo tar xvf sgx_rpm_local_repo.tgz
```

- e. Add the RPM local repository to your local repository list

```
sudo zypper addrepo /opt/intel/sgx_rpm_local_repo
sgx_rpm_local_repo
```

- f. Install all the latest packages using `'sudo zypper -no-gpg-checks2 install <package names>'`. Here is an example that installs all packages from the 2.11 release with DCAP 1.8.

```
sudo zypper -no-gpg-checks install libsgx-urts libsgx-launch
libsgx-epid libsgx-quote-ex libsgx-dcap-ql
```

- g. **(Optional)** To debug with `sgx-gdb`, install `*-debuginfo` packages to get the debug symbol. For example:

```
sudo zypper -no-gpg-checks install libsgx-aesm-ecdsa-plugin-
debuginfo libsgx-aesm-epid-plugin-debuginfo libsgx-aesm-
launch-plugin-debuginfo libsgx-aesm-pce-plugin-debuginfo
libsgx-aesm-quote-ex-plugin-debuginfo libsgx-dcap-default-
qpl-debuginfo libsgx-dcap-ql-debuginfo libsgx-dcap-quote-
verify-debuginfo libsgx-enclave-common-debuginfo libsgx-
epid-debuginfo libsgx-launch-debuginfo libsgx-pce-logic-
debuginfo libsgx-qe3-logic-debuginfo libsgx-quote-ex-
debuginfo libsgx-ra-network-debuginfo libsgx-ra-uefi-
debuginfo libsgx-urts-debuginfo sgx-aesm-service-debuginfo
sgx-pck-id-retrieval-tool-debuginfo sgx-ra-service-
debuginfo
```

---

<sup>2</sup> Since the local repo is not signed with GPG, ignore the GPG check when installing the packages.

- h. **(Optional)** To run an Intel® SGX application that loads an enclave requiring the Provision Key Access, the user needs to be added to the group of "sgx\_prv". This is true for any application that obtains a quote from the DCAP Quote Generation library for the purposes of remote attestation. Use the following command to add access for a user:

```
sudo usermod -aG sgx_prv <username>
```

- i. **(Optional)** To run enclave apps on distros with default "sgx" group, the user needs to be added to the group of "sgx ". Use the following command to add access for a user:

```
sudo usermod -aG sgx <username>
```

3. Set up the Intel® Software Guard Extensions Data Center Attestation Primitives (Intel® SGX DCAP) Provisioning Certificate Caching Service (PCCS) and Quote Provider Library (QPL). The PCCS and QPL work together to first cache DCAP attestation collateral information and then make the information available to the DCAP Quote Generation Library (libsgx-dcap-ql). These packages are provided as reference designs that users may employ. If you are running in a specific cloud, check with the CSP to see if it provides a CSP-specific QPL that accesses a custom provisioning certificate caching service in that CSP's cloud.

- a. [Optional] Setup the Provisioning Certificate Caching Service (PCCS):  
If the CSP already provides the caching service, you don't need to setup it by yourself. Please consult Cloud Service Provider about how to use their caching service. If you are not running in the cloud or there is no caching service available, please refer to [Appendix 2](#) for how to setup local caching service.

- b. Install the DCAP QPL package (If the CSP has a specific QPL then you need to install that one):

```
sudo zypper install --no-gpg-checks libsgx-dcap-default-ql
```

## Intel® SGX Application Developer

In addition to installing the Intel® Software Guard Extensions Platform Software (Intel® SGX PSW), you should also install the Intel® SGX Software Development Kit (Intel® SGX SDK) and the prerequisite software.

### **Ubuntu and Debian**

1. Install the prerequisite software. For more information about prerequisites, see *Install the Intel® SGX SDK: Prerequisites*: <https://github.com/intel/linux-sgx/blob/master/README.md>.  
Run the following command:

On Debian:

```
sudo apt-get install build-essential python3
```

```
sudo update-alternatives --install /usr/bin/python python  
/usr/bin/python3 1
```

On Ubuntu 20.04, Ubuntu 22.04 and Ubuntu 24.04:

```
sudo apt-get install build-essential python-is-python3
```

2. Download the Intel® SGX SDK and install it.

- a. Insert into the following command line the appropriate Linux distribution and its version, the Intel® SGX SDK version and build, and run the command:

```
wget - https://download.01.org/intel-sgx/latest/linux-latest/distro/  
<distro>/sgx_linux_x64_sdk_<version>.<build>.bin
```

- b. Adjust the file permissions:

```
chmod +x sgx_linux_x64_sdk_<version>.<build>.bin
```

- c. Start interactive setup by running the following command (run with sudo if necessary):

```
./sgx_linux_x64_sdk_<version>.bin
```

- d. When the question **Do you want to install in current directory? [yes/no]** appears, choose one of the following:

- If you want to install the components in the current directory, type **yes** and press **Enter**.
- If you want to provide another path for the installation, type **no** and press **Enter**.

Now the Intel® SGX SDK package is installed into the directory *<Your Input Location>/sgxsdk*. In this location you can also find a generated script `uninstall.sh`, which you can use to uninstall the Intel® SGX SDK.

- e. Run the following command to set all environment variables:

```
source <User Input Path>/sgxsdk/environment
```

- f. **(Optional)** Start non-interactive setup by running the following command (run with sudo if necessary):

```
./sgx_linux_x64_sdk_<version>.bin --prefix {SDK_INSTALL_PATH_PREFIX}
```

3. Install the appropriate developer packages `libsgx-enclave-common-dev`, `libsgx-dcap-ql-dev` and `libsgx-dcap-default-qp1-dev` with the following command:

```
sudo apt-get install libsgx-enclave-common-dev libsgx-dcap-ql-dev libsgx-  
dcap-default-qp1-dev tee_appraisal_tool
```

**Red Hat, CentOS and Anolis**

1. Install the prerequisite software. For more information about prerequisites, see *Install the Intel® SGX SDK: Prerequisites*: <https://github.com/intel/linux-sgx/blob/master/README.md>.  
Run the following command:

```
sudo yum groupinstall 'Development Tools'
```

2. Download the Intel® SGX SDK and install it.

- a. Insert into the following command line the appropriate Linux distribution and its version, the Intel® SGX SDK version and build, and run the command:

```
wget - https://download.01.org/intel-sgx/latest/linux-latest/distro/  
<distro>/sgx_linux_x64_sdk_<version>.<build>.bin
```

- b. Adjust the file permissions:

```
chmod +x sgx_linux_x64_sdk_<version>.<build>.bin
```

- c. Start interactive setup by running the following command (run with `sudo` if necessary):

```
./sgx_linux_x64_sdk_<version>.<build>.bin
```

- d. When the question **Do you want to install in current directory? [yes/no]** appears, choose one of the following:
  - o If you want to install the components in the current directory, type **yes** and press **Enter**.
  - o If you want to provide another path for the installation, type **no** and press **Enter**.

Now the Intel® SGX SDK package is installed into the directory *<Your Input Location>/sgxsdk*. In this location you can also find a generated script `uninstall.sh`, which you can use to uninstall the Intel® SGX SDK.

- e. Run the following command to set all environment variables:

```
source <User Input Path>/sgxsdk/environment
```

- f. **(Optional)** Start non-interactive setup by running the following command (run with `sudo` if necessary):

```
./sgx_linux_x64_sdk_<version>.bin --prefix {SDK_INSTALL_PATH_PREFIX}
```

3. Install the appropriate developer packages with the following command. **Note:** This command assumes you have already added the local SGX repo as instructed earlier in the “Red Hat, CentOS and Anolis” section under “Intel® SGX Application User”:

```
sudo yum install --nogpgcheck libsgx-enclave-common-devel libsgx-dcap-ql-  
devel libsgx-dcap-default-ql-devel libsgx-quote-ex-devel libsgx-launch-
```

```
devel libsgx-epid-devel libsgx-dcap-ql-devel libsgx-dcap-quote-verify-  
devel tee_appraisal_tool
```

### **SUSE Linux Enterprise Server**

1. Install the prerequisite software. For more information about prerequisites, see *Install the Intel® SGX SDK: Prerequisites*: <https://github.com/intel/linux-sgx/blob/master/README.md>.  
Run the following command:

```
sudo zypper install --type pattern devel_basis
```

2. Download the Intel® SGX SDK and install it.

- a. Insert into the following command line the appropriate Linux distribution and its version, the Intel® SGX SDK version and build, and run the command:

```
wget - https://download.01.org/intel-sgx/latest/linux-latest/distro/  
<distro>/sgx_linux_x64_sdk_<version>.<build>.bin
```

- b. Adjust the file permissions:

```
chmod +x sgx_linux_x64_sdk_<version>.<build>.bin
```

- c. Start interactive setup by running the following command (run with `sudo` if necessary):

```
./sgx_linux_x64_sdk_<version>.<build>.bin
```

- d. When the question **Do you want to install in current directory? [yes/no]** appears, choose one of the following:
  - If you want to install the components in the current directory, type **yes** and press **Enter**.
  - If you want to provide another path for the installation, type **no** and press **Enter**.

Now the Intel® SGX SDK package is installed into the directory *<Your Input Location>/sgxsdk*. In this location you can also find a generated script `uninstall.sh`, which you can use to uninstall the Intel® SGX SDK.

- e. Run the following command to set all environment variables:

```
source <User Input Path>/sgxsdk/environment
```

- f. **(Optional)** Start non-interactive setup by running the following command (run with `sudo` if necessary):

```
./sgx_linux_x64_sdk_<version>.bin --prefix {SDK_INSTALL_PATH_PREFIX}
```

3. Install the appropriate developer packages with the following command. **Note:** This command assumes you have already added the local SGX repo as instructed earlier in the “SUSE Linux Enterprise Server” section under “Intel® SGX Application User”:

```
sudo zypper install --nogpgcheck libsgx-enclave-common-devel libsgx-dcap-ql-devel libsgx-dcap-default-ql-devel libsgx-quote-ex-devel libsgx-launch-devel libsgx-epid-devel libsgx-dcap-ql-devel libsgx-dcap-quote-verify-devel tee_appraisal_tool
```

## Building the Intel® SGX Software Stack

### Intel® SGX - Platform Software and Software Development Kit

The source code for the Intel® Software Guard Extensions Platform Software (Intel® SGX PSW) and the Intel® Software Guard Extensions Software Development Kit (Intel® SGX SDK) is located in the following GitHub\* repository: <https://github.com/intel/linux-sgx>. To build and deploy the packages, follow the instructions in <https://github.com/intel/linux-sgx/blob/master/README.md>.

#### ***Prebuilt Binaries***

To run Intel® SGX enclaves on systems that use Legacy Launch Control and then to properly provision and use the Intel® EPID attestation, you must use certain enclaves called Architectural Enclaves or AEs, which are pre-built and signed. You can download these pre-built enclaves for the Intel® SGX Linux\* release from <https://download.01.org/intel-sgx/latest/linux-latest/>. The prebuilt enclaves are in a .tar file in the form `prebuilt_ae_<version>.tar.gz`

In addition, the Intel SDK provides prebuilt optimized libraries in the binary form. These libraries are provided in a .tar file in the form of `optimized_libs_<version>.tar.gz`.

Check the SHA256 hash of downloaded libraries using `SHA256SUM_prebuilt_<version>.cfg`.

### Intel® SGX Data Center Attestation Primitives

The source code for the Intel® Software Guard Extensions Data Center Attestation Primitives (Intel® SGX DCAP) is located in the following GitHub\* repository:

<https://github.com/intel/SGXDataCenterAttestationPrimitives>. To build and deploy the packages, follow the instructions in

<https://github.com/intel/SGXDataCenterAttestationPrimitives/blob/master/README.md>.

#### ***Prebuilt Binaries***

To use the Intel® SGX DCAP, you must also use certain enclaves that are pre-built and signed. This includes enclaves used by the Intel® SGX DCAP Quote Generation Library, which are located here:

<https://download.01.org/intel-sgx/latest/dcap-latest/linux/> in file `prebuilt_dcap_<version>.tar.gz`. For release notes and other details, see <https://download.01.org/intel-sgx/latest/dcap-latest/linux/docs/>

## Intel® Software Guard Extensions – Software Packages

---

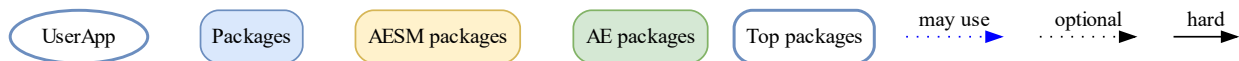
Figure 1 Intel(R) SGX PSW and Intel(R) SGX DCAP for Linux Package Structure details the set of packages that are provided in Intel® SGX DCAP beginning with Version 1.4 and Intel® SGX PSW beginning with version 2.8. The figure also displays required (hard) and optional dependencies.

While the figure contains many packages, the packages presented in blue, labeled in the key as *sgx library packages*, are of special interest to developers. Each of the *sgx library packages* presents a library that provides a function to an application developer.

Other packages, *aesm related packages*, provide special security functions related to the Architectural Enclave Service Manager (AESM). These packages are presented in yellow. The AESM is a daemon that provides special functions, such as quote generation or launch token generation, to Intel® SGX applications. The AESM has been modularized into a service, provided in package *sgx-aesm-service*, and a set of plug-ins where each plug-in provides a specific function.

Figure 2 Intel(R) SGX PSW and Intel(R) SGX DCAP for Linux Package Structure (TDX related) details the set of packages for Intel® TDX that are provided in Intel® SGX DCAP beginning with Version 1.14 and Intel® SGX PSW beginning with version 2.17. The figure also displays required (hard) and optional dependencies.

In Intel TDX®, the packages are separated in 2 domains. Most of them should be installed on host (or in a dedicated SGX VM or a docker container), like *tdx-qgs*, *libsgx-dcap-default-qpl*, *sgx-dcap-pccs*, etc. In the TD guest, applications that need to get TD report or TD quote depend on *libtdx-attest* package. It's the only package need to be installed in the TD guest.



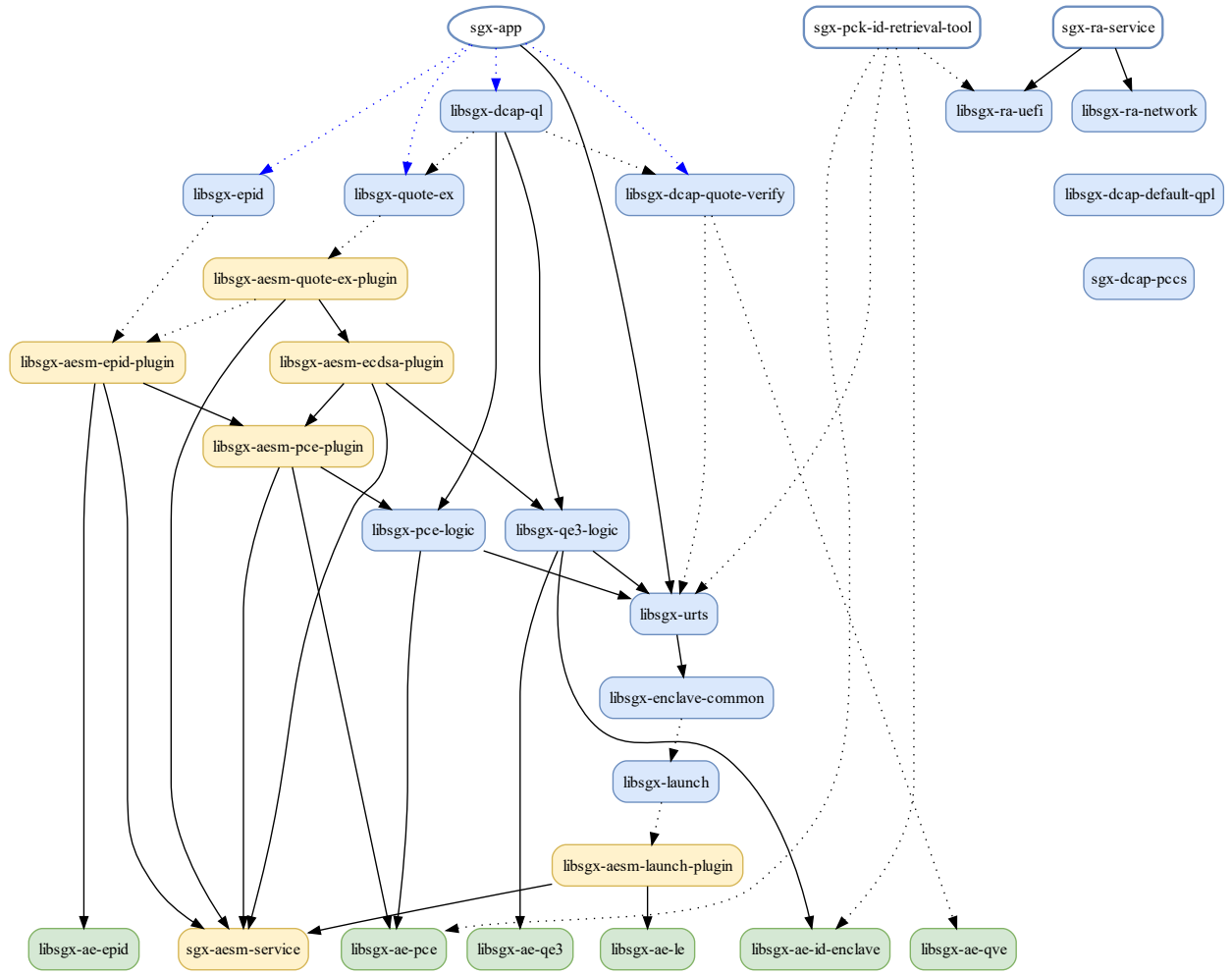


Figure 1 Intel(R) SGX PSW and Intel(R) SGX DCAP for Linux Package Structure



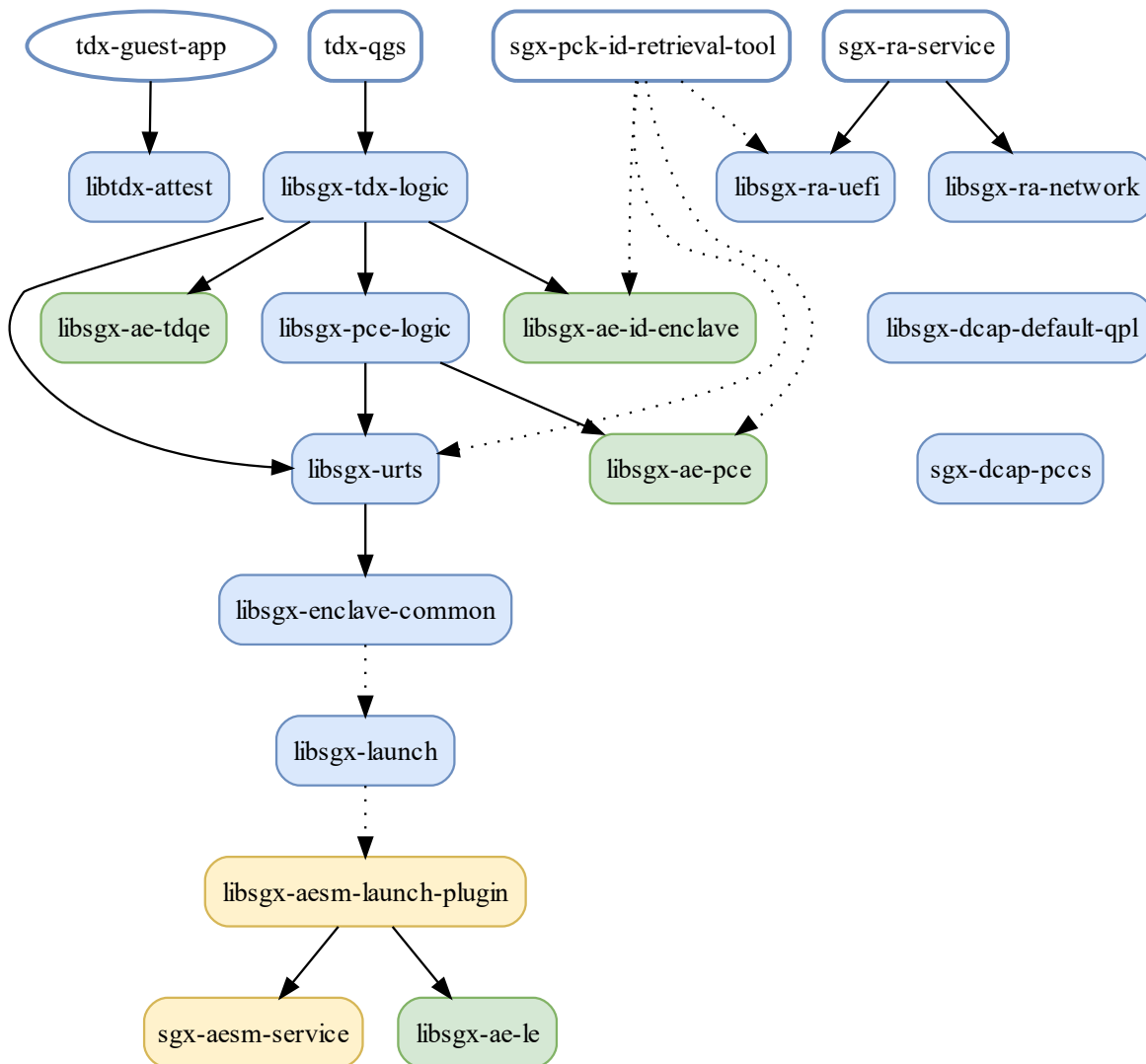


Figure 2 Intel(R) SGX PSW and Intel(R) SGX DCAP for Linux Package Structure (TDX related)

## Access to AESM Plug-ins

For an application to use a specific AESM function, the application installation must ensure that the function’s plug-in is installed. This can be done in one of two methods:

1. Set a direct dependency on the specified service’s plug-in package. This installs the service in the environment where the application is installed.
2. Ensure that the plug-in is installed in an accessible environment where its service can be accessed. Each function that the AESM plug-ins provide can be accessed via a Unix Domain Socket with a hardcoded port. The plug-in may be installed in one environment, for example a container, and the Unix Domain Socket port may be relayed to an IP socket via a utility such as

[SOCAT](#) so that it can be accessed in another environment (using a corresponding [SOCAT](#) configuration to relay the bytestream to the Unix Domain Socket).

Each AESM plug-in package depends on the AESM Service (*sgx-aesm-service*) itself and also upon one or more Architectural Enclave (AE) packages denoted in green. AE packages provide enclaves themselves. These enclaves, such as *libsgx-ae-pce*, are provided in their own packages to allow them to be updated (some are in the TCB of the Intel® SGX application due to the service they provide) and be signed by a trusted entity (the signer of an AE must be trusted by the party relying on the AE service).

It should be noted that a packages *libsgx-dcap-default-qpl* and *sgx-dcap-pccs* are not presented with any direct dependencies. These are described in a later section.

## Detailed Description of Packages

The best description of the individual components relates to what you need to do in an application or a run time, or how you may configure a VM or platform.

Each package and its dependencies are listed in alphabetical order in Table 1: Intel(R) Linux SGX Software Packages.

Package Name	Type	Function	Direct SGX Package Dependency(s)	Developer Files
<i>libsgx-ae-epid</i>	AE	Intel® EPID Quoting Enclave	None	EDL provided in release package
<i>libsgx-ae-le</i>	AE	Legacy Launch Enclave	None	EDL provided in release package
<i>libsgx-ae-pce</i>	AE	PCE Enclave	None	EDL provided in release package
<i>libsgx-ae-qe3</i>	AE	ECDSA Quoting Enclave	None	EDL provided in release package
<i>libsgx-ae-qve</i>	AE	ECDSA Quote Verification Enclave	None	EDL provided in release package
<i>libsgx-ae-id-enclave</i>	AE	ID Enclave	None	EDL provided in release package
<i>libsgx-ae-tdqe</i>	AE	TDX Quoting Enclave	None	EDL provided in release package
<i>libsgx-aesm-ecdsa-plugin</i>	Plug-in	Plug-in to provide ECDSA Quotes	<i>libsgx-qe3-logic</i> <i>libsgx-aesm-pce-plugin</i> <i>sgx-aesm-service</i>	
<i>libsgx-aesm-epid-plugin</i>	Plug-in	Plug-in to provide Intel® EPID Quotes	<i>libsgx-ae-epid</i> <i>libsgx-aesm-pce-plugin</i> <i>sgx-aesm-service</i>	
<i>libsgx-aesm-launch-plugin</i>	Plug-in	Plug-in to provide Legacy Launch Tokens	<i>libsgx-ae-le</i> <i>sgx-aesm-service</i>	
<i>libsgx-aesm-pce-plugin</i>	Plug-in	Plug-in to provide PCE Signing	<i>Libsgx-pce-logic</i> <i>libsgx-ae-pce</i> <i>sgx-aesm-service</i>	
<i>libsgx-aesm-quote-ex-plugin</i>	Plug-in	Plug-in to provide universal quoting	<i>libsgx-aesm-ecdsa-plugin</i> <i>libsgx-aesm-epid-plugin</i> (opt.) <i>sgx-aesm-service</i>	
<i>libsgx-dcap-default-qpl</i>	Lib	Default Quote Provider Library (QPL)	None	<i>libsgx-dcap-default-qpl-dev</i>

## Intel® Software Guard Extensions - Software Installation Guide for Linux\* OS

libsgx-decap_default-ql-dev*	Dev Files	Developer Files for Default Quote Provider Library	libsgx-dcap-default-ql	
libsgx-dcap-ql	Lib	Library that provides ECDSA Quotes	libsgx-pce-logic libsgx-qe3-logic libsgx-quote-ex(optional) libsgx-dcap-quote-verify(opt.)	libsgx-dcap-default-ql-dev
libsgx-dcap-ql-dev*	Dev Files	Developer Files for libsgx-decap-ql	libsgx-headers libsgx-dcap-ql	
libsgx-dcap-quote-verify	Lib	Quote Verification Library	libsgx-urts(optional) libsgx-ae-qve(optional)	libsgx-dcap-quote-verify-dev
libsgx-dcap-quote-verify-dev*	Dev Files	Developer files for Quote Verification Library	libsgx-headers libsgx-dcap-quote-verify	
libsgx-pce-logic	Lib	Library that provides PCE logic wrapper	libsgx-urts libsgx-ae-pce	
libsgx-qe3-logic	Lib	Library that provides ECDSA QE logic wrapper	libsgx-urts libsgx-ae-qe3 libsgx-ae-id-enclave	
libsgx-tdx-logic	Lib	Library that provides TD QE logic wrapper	libsgx-urts libsgx-pce-logic libsgx-ae-tdqe libsgx-ae-id-enclave	libsgx-tdx-logic-dev
libsgx-tdx-logic-dev*	Dev Files	Developer Files for TD QE logic wrapper Library	libsgx-tdx-logic	
libsgx-enclave-common	Lib	Library that presents a common interface for loading SGX enclaves	libsgx-launch (optional)	libsgx-enclave-common-dev
libsgx-enclave-common-dev*	Dev Files	Developer files for Enclave Common	libsgx-headers libsgx-enclave-common	
libsgx-epid	Lib	Library that provides Intel® EPID quotes	libsgx-aesm-epid-plugin (optional)	libsgx-epid-dev
libsgx-epid-dev*	Dev Files	Developer files for libsgx-epid	libsgx-headers libsgx-epid	
libsgx-launch	Lib	Library that provides legacy launch tokens	libsgx-aesm-launch-plugin (optional)	libsgx-launch-dev
libsgx-launch-dev*	Dev Files	Developer files for libsgx-launch	libsgx-headers libsgx-launch	
libsgx-quote-ex	Lib	Library that provides agnostic quoting (current Intel® EPID or ECDSA)	libsgx-aesm-quote-ex-plugin (optional)	libsgx-quote-ex-dev
libsgx-quote-ex-dev*	Dev Files	Developer files for libsgx-quote-ex	libsgx-headers libsgx-quote-ex	
libsgx-uae-service	Lib (deprecated)	Library that wraps libsgx-epid, libsgx-launch, and other deprecated features. Provided to support legacy apps.	libsgx-epid libsgx-launch libsgx-quote-ex	
libsgx-urts	Lib	Library that provides uRTS features to load/manage Intel® SGX Enclaves	libsgx-enclave-common	Header files provided in SDK: sgx_linux_x64_sdk_*.bin
sgx-aesm-service	Serv	AESM Service that runs AESM plug-ins	None	None
tdx-qgs	Serv	TD Quoting Generation Service	libsgx-tdx-logic	
sgx-dcap-pccs	Serv	PCK Cert. Caching Service	None	
libtdx-attest	Lib	Trust Domain Extensions Attestation library	None	libtdx-attest-dev

libtdx-attest-dev*	Dev Files	Developer files for libtdx-attest	libtdx-attest	
--------------------	-----------	-----------------------------------	---------------	--

Table 1: Intel(R) Linux SGX Software Packages (\*Package names may vary by OS)

## FIPS 140-3 Certifiable mode

Some SGX software components are built with OpenSSL3 static library. To make them FIPS 140-3 Certifiable, you need to provide openssl.cnf, fipsmodule.cnf and fips.so for your system.

1. Make sure you have following sections in your openssl.cnf, then `export OPENSSL_CONF=/path/to/your/openssl.cnf`

```
# For FIPS
# Optionally include a file that is generated by the OpenSSL fipsinstall
# application. This file contains configuration data required by the OpenSSL
# fips provider. It contains a named section e.g. [fips_sect] which is
# referenced from the [provider_sect] below.
# Refer to the OpenSSL security policy for more information.
.include /path/to/your/fipsmodule.cnf
```

```
[openssl_init]
providers = provider_sect

# List of providers to load
[provider_sect]
# default = default_sect
# The fips section name should match the section name inside the
# included fipsmodule.cnf.
fips = fips_sect
base = base_sect
```

2. If you have fips.so in `/path/to/your/fips.so`, you need to:

```
export OPENSSL_MODULES=/path/to/your/
```

Then SGX related libraries and tools will load fips.so and use the FIPS 140-3 Certifiable APIs.

Please note: Since AESM is running as a daemon. You need to change /usr/lib/systemd/system/aesmd.service to add 2 lines as below (using the example path aforementioned):

*[Unit]*

*Description=Intel(R) Architectural Enclave Service Manager*

*After=syslog.target network.target auditd.service*

*After=remount-dev-exec.service*

*Wants=remount-dev-exec.service*

*[Service]*

*User=aesmd*

*Type=forking*

*Environment=NAME=aesm\_service*

*Environment=AESM\_PATH=/opt/intel/sgx-aesm-service/aesm*

*Environment=LD\_LIBRARY\_PATH=/opt/intel/sgx-aesm-service/aesm*

***Environment=OPENSSL\_MODULES=/path/to/your/***

***Environment=OPENSSL\_CONF=/path/to/your/openssl.cnf***

*WorkingDirectory=/opt/intel/sgx-aesm-service/aesm*

*PermissionsStartOnly=true*

*ExecStartPre=/opt/intel/sgx-aesm-service/aesm/linksgx.sh*

*ExecStartPre=/bin/mkdir -p /var/run/aesmd/*

*ExecStartPre=/bin/chown -R aesmd:aesmd /var/run/aesmd/*

*ExecStartPre=/bin/chmod 0755 /var/run/aesmd/*

*ExecStartPre=/bin/chown -R aesmd:aesmd /var/opt/aesmd/*

*ExecStartPre=/bin/chmod 0750 /var/opt/aesmd/*

*ExecStart=/opt/intel/sgx-aesm-service/aesm/aesm\_service*

*InaccessibleDirectories=/home*

*ExecReload=/bin/kill -SIGHUP \$MAINPID*

*Restart=on-failure*

*RestartSec=15s*

*DevicePolicy=closed*

```
DeviceAllow=/dev/isgx rw
DeviceAllow=/dev/sgx rw
DeviceAllow=/dev/sgx/enclave rw
DeviceAllow=/dev/sgx/provision rw
DeviceAllow=/dev/sgx_enclave rw
DeviceAllow=/dev/sgx_provision rw
[Install]
WantedBy=multi-user.target
```

Then

```
sudo systemctl daemon-reload
sudo service aesmd restart
```

For detailed information, you can refer

[https://www.openssl.org/docs/manmaster/man7/fips\\_module.html](https://www.openssl.org/docs/manmaster/man7/fips_module.html)

## Summary of Use Cases and Required Packages

The packages that are required, either as application package dependencies or as platform dependencies, depend on the use case of the application and the platform environment.

### Load Custom Intel® SGX Enclaves

The loading process for the Intel® SGX enclaves with a custom configuration (such as enclave produced from the Open Enclave SDK) should use the *libsgx-enclave-common* library.

- Package: *libsgx-enclave-common*
- Documentation: [Intel SGX Enclave Common Loader API Reference.pdf](#)
- Comments:
  - This library is typically used by enclave run times such as Open Enclave Loader or Intel® SGX uRTS
  - If legacy launch is used on the platform, *libsgx-enclave-common* needs the package *libsgx-launch* and its dependent packages

### Load Intel® SGX Enclaves

Enclaves that are produced using the Intel® SGX SDK are loaded and managed with the *libsgx-urts* library.

- Package: *libsgx-urts*

- Documentation: *Intel\_SGX\_Developer\_Reference\_Linux\_<version>\_Open\_Source.pdf* from <https://download.01.org/intel-sgx/latest/linux-latest/>
- Comments:
  - *libsgx-urts* depends on *libsgx-enclave-common*
  - To use in your application, install the Intel® SGX SDK for Linux from <https://01.org/intel-software-guard-extensions/downloads>

## Use Intel® SGX DCAP Quoting

Applications that use DCAP Quoting (also referred to as ECDSA Quoting) should use the *libsgx-dcap-ql* library.

- Package: *libsgx-dcap-ql*
- Documentation: *Intel\_SGX\_Developer\_Reference\_Linux\_<version>\_Open\_Source.pdf* from <https://download.01.org/intel-sgx/latest/linux-latest/>
- Comments:
  - *libsgx-dcap-ql* directly depends on:
    - *libsgx-urts* (dependency is not shown on diagram)
    - *libsgx-ae-qve*
    - *libsgx-ae-qe3*
    - *libsgx-ae-pce*

## Use Intel® EPID Quoting

Applications that use Intel® EPID Quoting should use the *libsgx-epid* library.

- Package: *libsgx-epid*
- Documentation: *Intel\_SGX\_Developer\_Reference\_Linux\_<version>\_Open\_Source.pdf* from <https://download.01.org/intel-sgx/latest/linux-latest/>
- Comments:
  - *libsgx-epid* requires access to *libsgx-aesm-epid-plugin*. This can be a hard dependency in the application package, or it can be provided independently on the platform. See Access to AESM Plug-ins for details

## Use Universal Quoting

Applications that use Universal Quoting should use the *libsgx-quote-ex* library.

- Package: *libsgx-quote-ex*
- Documentation: *Intel\_SGX\_Developer\_Reference\_Linux\_<version>\_Open\_Source.pdf* from <https://download.01.org/intel-sgx/latest/linux-latest/>

- Comments:
  - *libsgx-quote-ex* requires access to *libsgx-quote-ex-plugin*. This can be a hard dependency in the application package, or it can be provided independently on the platform. See Access to AESM Plug-ins for details

## Use TDX Attestation in a TD guest

Applications that use TDX attestation should use the *libtdx-attest* library.

- Package: *libtdx-attest*

## Provide TD Quoting Service

To provide TD Quoting service on TDX enabled host should install the *tdx-qgs* service.

- Package: *tdx-qgs*
- Comments:
  - *tdx-qgs* requires *libsgx-tdx-logic* as a hard dependency. It also requires *libsgx-dcap-default-qpl* in runtime to get platform information, while *libsgx-dcap-default-qpl* is not a dependency of *tdx-qgs*. It's up to system administrator to install the default qpl provided by Intel or their own one.

## Platform Uses Legacy Launch

Legacy Launch refers to a platform configuration where a launch token must be provided by an Intel-signed Launch Enclave. Whether a platform (in this case platform can be a VM or a bare-metal platform) requires legacy launch is determined by two factors:

- The platform HW and configuration: All platforms configured with Intel® SGX support have the ability to support legacy launch; however, only platforms supporting Intel® SGX Launch Configuration (Detailed in CPUID[7].ECX[SGX\_LC] where SGX\_LC is bit 30.) and configured for Intel® SGX Launch Control (Specified in IA32\_FEATURE\_CONTROL MSR bit 17, which is configured by the BIOS) are capable of running without Legacy Launch. See the Intel® Software Developers Manual for more information.
- The Intel® SGX Support in the Linux Kernel: Intel® SGX can be supported in the Linux Kernel via a patch to the kernel or an installed driver. In addition, there are currently two supported drivers for Intel® SGX. Thus, there are three options to configure a platform for Intel® SGX Launch:
  - In-Kernel Driver: the kernel has been patched to support Intel® SGX. This may only run on platforms that are capable of supporting Intel® SGX Launch Configuration and have the feature enabled.
  - DCAP Driver: the DCAP Driver is a Linux kernel mode driver that mimics the In-kernel Driver solution. As with the In-kernel Driver, the DCAP driver only runs on platforms that are capable of supporting Intel® SGX Launch Configuration and have the feature enabled.



- Out-of-Tree Driver: The Out-of-Tree Driver name is a misnomer as the DCAP Driver is also provided out-of-tree; however, this driver can run on all platforms whether Intel® SGX Launch Configuration is supported or enabled. Thus, it supports Legacy Launch. It has an interface that requires a Launch Token to be provided when initializing an enclave. For more information on the Intel® SGX Launch Token, see the Intel® Software Developers Manual.

If a platform is configured with the Out-of-Tree driver, it should also install *libsgx-launch* and *libsgx-aesm-launch-plugin*. Library *libsgx-enclave-common* automatically detects the installed driver and adapts to use the *libsgx-launch* library to obtain a launch token

- Package: *libsgx-launch*
- Documentation: See Intel\_SGX\_Developer\_Reference\_Linux\_<version> in <https://download.01.org/intel-sgx/latest/linux-latest/docs/>
- Comments:
  - When Legacy Launch is used, release version enclaves must be signed with a key that has been adopted into the Intel® SGX Launch Policy List. For information on how to register a key, see <https://software.intel.com/en-us/sgx/request-license>

## Platform uses Intel Reference Provisioning Certificate Caching Service

ECDSA Quoting and Quote Verification (used in DCAP) require collateral information provided by Intel® Services. This information should be cached within a cloud environment using a caching service. Intel provides a reference Provisioning Certificate Caching Service (PCCS) in package *sgx-dcap-pccs*. In addition, it provides a reference library, the Quote Provider Library or QPL, in package *libsgx-dcap-default-qpl* that is used by *libsgx-dcap-ql* and *libsgx-aesm-ecdsa-plugin* to obtain collateral from the PCCS. The configuration of the QPL and PCCS is beyond the scope of this document.

- Package: *sgx-dcap-pccs* and *libsgx-dcap-default-qpl*
- Documentation: See SGX\_DCAP\_Caching\_Service\_Design\_Guide\_<version>.pdf and Intel\_SGX\_ECDSA\_QuoteLibReference\_DCAP\_API.pdf in <https://download.01.org/intel-sgx/latest/dcap-latest/linux/docs/>
- Comments:
  - Cloud Service Providers are likely to deploy their own Quote Provider Library and Provisioning Certificate Caching Service. In this case, the platform should be provisioned with the CSPs specific Software.

## Intel® SGX Software Development Kit for Linux\* OS

The Intel® Software Guard Extensions Software Development Kit (Intel® SGX SDK) for Linux\* OS provides libraries, tools, reference code, and documentation that help you code, build, and sign Intel® SGX enclaves and the applications that host Intel® SGX enclaves.

The Intel® SGX SDK installation is provided as a binary file:

- Location: <https://download.01.org/intel-sgx/> linux-<version>/<OS><OS version>
- Filename: `sgx_linux_x64_sdk_<version>.<build>.bin`

## Dependencies

- `build-essential`
- `python3`
- `libsgx-urts` and other SGX packages (required by sample code)

See *Install the Intel® SGX SDK: Prerequisites* at <https://github.com/intel/linux-sgx/blob/master/README.md#prerequisites-1>.

## Source

Source code for the Intel® SGX SDK for Linux\* OS is located on GitHub\*:

- Source code: <https://github.com/intel/linux-sgx>.
- Build instructions: <https://github.com/intel/linux-sgx/blob/master/README.md>. This document contains detailed instructions on platform configuration and build procedures for the Intel® SGX SDK and the Intel® SGX PSW for Linux\* OS.
- Build dependencies: <https://github.com/intel/linux-sgx/blob/master/README.md>. This document defines installation prerequisites and build dependencies.

## Appendix 1: Advanced Configuration Topics

---

This appendix describes several advanced configuration topics:

- Access to ECDSA Quote Generation using the DCAP Quoting Library: to obtain a Quote for ECDSA based attestation, there are two methods. These methods require specific configuration:
  - Configuration of Out-of-Process ECDSA Quote Generation: to configure an application to use the AESM Service to acquire ECDSA quotes.
  - Launching an enclave with the provision bit set: In-Process ECDSA Quote Generation
- Configuration of AESM Proxy Service: to configure an http proxy server for the AESM Service to use.

### Access to ECDSA Quote Generation using the DCAP Quoting Library

For an enclave to attest to a remote entity, it must obtain a report of itself and then get that report signed into a “quote” by a special quoting enclave (aka “QE3”) on the platform. Running the QE3 enclave in an application requires the user to have special privilege on the system. Some processes may not have this privilege. Thus, the Intel® DCAP Quoting Library offers two options to obtain an ECDSA based quote:

- In-Process ECDSA Quote Generation: the DCAP Quoting Library will load the QE3 enclave and obtain the quote with a direct call to the enclave.
- Out-of-Process ECDSA Quote Generation: in this case, the DCAP Quoting Library makes an out-of-process call to the AESM Service to get the quote.

Each of these options has specific requirements on the configuration of the software and environment on the platform or on the privilege of the user application.

#### In-Process ECDSA Quote Generation<sup>3</sup>

When using the DCAP Quoting Library, the default configuration is for the library to directly load the enclaves required to generate an ECDSA based quote. This may cause issues if you use DCAP Driver (version 1.41 and later) or in-kernel SGX Linux Driver (Linux Kernel 5.11 and later) because both require you to have a specific access privilege in order to launch an enclave, the QE3 enclave, capable of signing a quote. You must have permission to launch an enclave with the Provision Bit set.

---

<sup>3</sup> This information is partially reproduced from <https://github.com/intel/SGXDataCenterAttestationPrimitives/blob/master/driver/linux/README.md>

This section provides details on how to configure a user/process to support in-process ECDSA Quote Generation where the QE3 enclave will be loaded in-process.

An enclave may set the provision bit in its attributes to be able to request provision key. Acquiring provision key may have privacy implications and thus the permission to acquire the key should be limited to privileged users. Such enclaves are referred to as Provisioning Enclaves below.

For applications loading provisioning enclaves, the platform owner (administrator) must grant provisioning access to the app process as described below.<sup>4</sup>

### **Driver Settings**

The DCAP Driver installation process described above creates two devices on the platform and configures these devices with the following permissions:

```
crw-rw-rw- root root      /dev/sgx/enclave
crw-rw---- root sgx_prv  /dev/sgx/provision
```

**Note:** The DCAP driver, or libsgx-enclave-common, sgx-aesm (see <https://01.org/intel-software-guard-extensions/downloads>) in PSW release 2.13 or higher automatically copies the udev rules and runs `udevadm trigger` to activate the rules so that the permissions are set as above.

This configuration enables every user to launch an enclave, but only those who are members of the `sgx_prv` group can launch enclaves with the provision bit set. Failing to set these permissions may prevent processes that are not running under root privilege from launching a Provisioning Enclave.

If In-Kernel Driver is used with PSW 2.10–2.12, then you need to manually set the udev rules and permissions. To do this, run following commands as root:

```
# cat > /etc/udev/rules.d/90-sgx-v40.rules <<EOF
SUBSYSTEM=="misc",KERNEL=="sgx_enclave",MODE="0666",SYMLINK+="sgx/enclave"
SUBSYSTEM=="misc",KERNEL=="sgx_provision",GROUP="sgx_prv",MODE="0660",SYMLINK
+="sgx/provision"
EOF
# udevadm trigger
```

PSW releases lower than 2.10 cannot be used with In-Kernel Driver.

---

<sup>4</sup>**Note for Intel Signed Provisioning Enclaves:** The Intel(R) SGX driver before V1.41 allows Intel's provisioning enclaves to be launched without any additional permissions. But the special treatment for Intel signed enclaves is removed from the DCAP driver starting from the V1.41 release. This is to align the DCAP Driver with the In-Kernel driver. If you upgrade driver from versions higher than 1.41 or switch to future mainline kernel with SGX support, make sure apps that load Intel-signed provisioning enclaves have the right permissions as described below.

### ***Process Permissions and Flow***

A process that launches a Provisioning Enclave is required to use the SET\_ATTRIBUTE IOCTL before the INIT\_ENCLAVE IOCTL to notify the driver that the enclave being launched requires provision key access. The SET\_ATTRIBUTE IOCTL input is a file handle to /dev/sgx/provision, which fails to open if the process does not have the required permission. To summarize, the following flow is required by the platform admin and a process that requires provision key access:

- Software installation flow:
  - Add the user running the process to the sgx\_prv group:

```
sudo usermod -a -G sgx_prv <user name>
```
- Enclave launch flow:
  - Create the enclave via the CREATE\_ENCLAVE IOCTL
  - Open a handle to /dev/sgx/provision
  - Issue the SET\_ATTRIBUTE IOCTL with the handle as a parameter
  - Continue the load and initialization of the enclave

**Note:** The Enclave Common Loader library is following the above flow and launching enclave based on it, failure to grant correct access to the launching process will cause a failure in the enclave initialization.

### **Out-of-Process ECDSA Quote Generation**

When using the DCAP Quoting Library, the default configuration is for the library to directly load the enclaves required to generate an ECDSA based quote in-process. An alternate method is to instruct the DCAP Quoting Library to use Universal Quoting (using the *libsgx-quote-ex* library), which makes a remote process call to the AESM Service to obtain a quote from the AESM Service. To do this:

- **Ensure that the proper packages are installed to support Out-of-Process ECDSA Quote Generation:** to do this, see Use Universal Quoting for the proper packages to install.
- **Create an environment variable named SGX\_AESM\_ADDR:** The existence of an environment variable named SGX\_AESM\_ADDR will instruct the DCAP Quoting Library to use Universal Quoting, which will access the AESM Service to obtain the quote. The AESM Service is preconfigured to run with a *sgx\_prv* privilege.

There are several ways to configure the SGX\_AESM\_ADDR environment variable to support out-of-process quote generation:

1. Add the environment variable to the command line when running the application:

```
SGX_AESM_ADDR=1 <app_name>
```

This only configures the out-of-process quote generation for the application being executed.

2. Add the environment variable to file `/etc/environment`. Add the line `SGX_AESM_ADDR=1` to the file. This configures the out-of-process quote generation for the whole system.

## Configuration of AESM Proxy Service

The Intel® SGX Software includes a service application, the AESM Service, which provides functionality to applications on the platform. Many of the library packages are installed as plug-ins to the AESM service and thus provide their functionality to the system while running within the AESM service

The AESM service executable is installed to the directory:

```
/opt/intel/sgx-aesm-service
```

The installer also configures the AESM service to run as a system daemon, which starts with the user ID `aesmd`. The default home directory of the AESM service is `/var/opt/aesmd`.

To perform certain functions such as EPID provisioning, the AESM need internet access. If your network is using a proxy service, you may need to configure the proxy for the ASEM. For instructions on setting up the proxy, refer to the file `/etc/aesmd.conf`.

## Start AESM without systemd and syslog, and other command line options

For cloud native (e.g. k8s) deployment, containers usually do not have `systemd` and `syslog`. You can start the AESM service program directly from shell command:

```
./aesm_service --no-daemon
```

You can also output logs to `stdout/stderr`:

```
./aesm_service --no-daemon --no-syslog
```

You can specify what attestation types are supported. If an attestation type is specified but AESM fails to load the corresponding modules, AESM will stop running. Currently only EPID and ECDSA are supported:

```
./aesm_service --supported_attestation_types=[EPID] [,ECDSA]
```

## Appendix 2: How to setup Provisioning Certificate Caching Service (PCCS) on local machine

---

1) Install `node.js`

Please check <https://github.com/nodesource/distributions> Make sure the correct node version was installed (Supported versions are 18.17.0 to 18.19.1, 20.0.0 to 20.11.1, or 21.0.0 to 21.5.0):

```
node -version
```

To install a specific version of Node.js, please refer to the instructions tailored for your operating system on this wiki page:

[How to select the Node.js version to install · nodesource/distributions Wiki \(github.com\)](https://github.com/nodesource/distributions/wiki/How-to-select-the-node-version-to-install)

2) Install the PCCS DCAP package:

Ubuntu:

```
sudo apt-get install sgx-dcap-pccs
```

RedHat:

```
sudo yum install --nogpgcheck sgx-dcap-pccs
```

Note: If you are behind a proxy and you get a network timeout error, you need to set the proxy for npm:

```
sudo npm config set proxy "http://proxy-server:port"
```

By default, the Debian package installer guides you through the configuration of the PCCS service. But on RedHat you need to execute the install script manually:

**[RedHat only]** `sudo -u pccs /opt/intel/sgx-dcap-pccs/install.sh`

If you want to better understand the options during installation or change the configuration after installation has been completed, see

<https://github.com/intel/SGXDataCenterAttestationPrimitives/tree/master/QuoteGeneration/pccs#configuration-file-configdefaultjson->

3) Start the PCCS service:

```
sudo systemctl start pccs
```

4) Run the following command to make sure the PCCS service is working correctly (assume the service is running on localhost with an insecure certificate):

```
curl -k -G "https://localhost:8081/sgx/certification/v4/rootcacrl"
```

The root CA CRL should be retrieved successfully.

**Important:**

1. If you are using insecure certificate for the PCCS service, [Deprecated] set `"USE_SECURE_CERT"=FALSE` in `/etc/sgx_default_qcnl.conf` after installing the `libsgx-dcap-default-qpl` package.

You need to set “use\_secure\_cert”:false in configuration file. For how to configure Default Quote Provider Library, please check

<https://github.com/intel/SGXDataCenterAttestationPrimitives/blob/master/QuoteGeneration/qpl/README.md>

2. It is recommended to backup the old database before you install the new version PCCS. The new PCCS will upgrade the database automatically
3. After making this change, you need to restart the PCCS with this command:  
`sudo systemctl restart pccs`



## Disclaimer and Legal Information

---

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.

The products and services described may contain defects or errors known as errata which may cause deviations from published specifications. Current characterized errata are available on request.

Intel technologies features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at Intel.com, or from the OEM or retailer.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting [www.intel.com/design/literature.htm](http://www.intel.com/design/literature.htm).

Intel, the Intel logo, Xeon, and Xeon Phi are trademarks of Intel Corporation in the U.S. and/or other countries.

### Optimization Notice

Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Notice revision #20110804

\* Other names and brands may be claimed as the property of others.

© Intel Corporation.

This software and the related documents are Intel copyrighted materials, and your use of them is governed by the express license under which they were provided to you (**License**). Unless the License

provides otherwise, you may not use, modify, copy, publish, distribute, disclose or transmit this software or the related documents without Intel's prior written permission.

This software and the related documents are provided as is, with no express or implied warranties, other than those that are expressly stated in the License.