

Implementing Intel SGX on Linux

Planning / Implementation

In the current landscape of pervasive security threats, ensuring the confidentiality and integrity of sensitive information is of paramount importance. Traditional software security measures often fall short in protecting data from sophisticated attacks that exploit system vulnerabilities. Intel SGX addresses this challenge by providing hardware-based security mechanisms that isolate sensitive computations and data from the rest of the system.

SGX introduces the concept of enclaves-secure regions of memory that are protected even from privileged system software. These enclaves ensure that sensitive data and code can be executed in a secure environment, free from tampering and eavesdropping. This technology is used in the scenarios of where secure data processing and confidentiality are crucial, such as in financial transactions, healthcare data processing, and confidential communications.

This paper has the following topics:

1. High-level introduction to Intel SGX, its purpose, importance, and the issues it aims to solve.
2. SGX functionalities, architecture, and insights.
3. How to enable Intel SGX on Lenovo systems, overview of configuration on UEFI settings and operating system settings.
4. How to build SGX test environment and validate whether SGX is correctly enabled.
5. SGX support information.
6. Troubleshooting when building environment and testing SGX.

For our test environment, we used RHEL 9.4 on a ThinkSystem SR630 V4 server with Intel Xeon 6787P processors.

Understanding Intel SGX

Intel SGX is designed to enhance data protection and improve the security of application code. This hardware-based CPU defense mechanism allows applications to operate within a private memory space, pre-allocated to specific areas of RAM. SGX introduces the concept of an *Enclave*, a secure environment with its own privileged mode of execution. This ensures that even high-privilege software, such as the operating system or BIOS, cannot access the data safeguarded by SGX.

When developing an Intel SGX application, programmers can decide which parts of the code should be placed inside an enclave and which should remain outside. They can designate certain sections as the "trusted part," responsible for handling sensitive data and private code segments, while the rest of the code operates as the "untrusted part". The untrusted part functions are similar to non-SGX applications and can be accessed by other high-privilege software.

The following is the complete workflow of an SGX application.

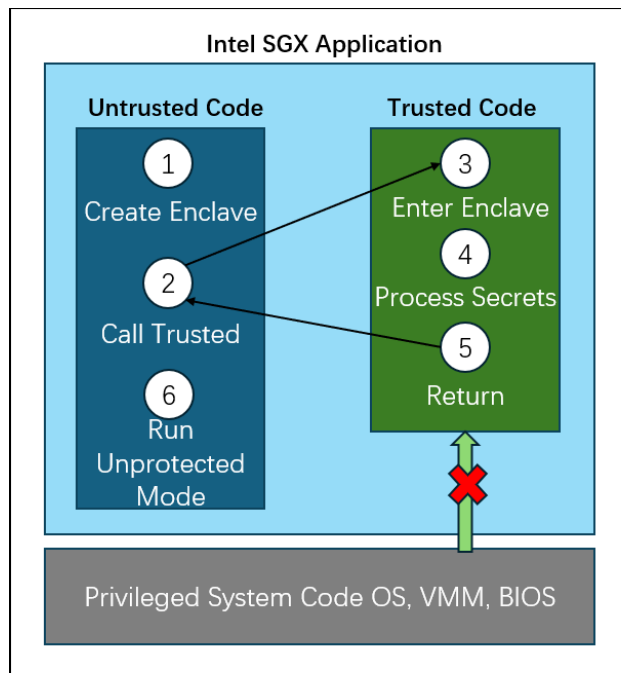


Figure 1. Workflow of an SGX application

Building an SGX application requires developers to modify their code. When the application is running, the following workflow is executed:

1. Application runs and uses "sgx_create_enclave" to create a new enclave region. This enclave region is used to store the code segments and data.
2. Application uses "Ecalls" to execute the trusted function.
3. Application execution is transitioned to the enclave mode.
4. Application execution, code segments and data are allowed to be accessed in the enclave memory. Any external access to the enclave memory is denied.
5. The application uses "Ocalls" for function returns, and the enclave data remains within the trusted memory.
6. Continue to run the unprotected code.

SGX technology insights

The software isolation environment provided by SGX is mainly achieved through a special memory management mechanism. The following is an introduction to SGX-specific terminology.

1. DRAM (Dynamic Random-Access Memory): CPU memory.
2. PRM (Processor Reserved Memory): The pre-reserved memory on DRAM used by SGX.
3. EPC (Enclave Page Cache): The PRM memory organized in pages, each page is 4k generally.
4. EPCM (Enclave Page Cache Map): record allocations for each EPC page, permission recording and page access control.

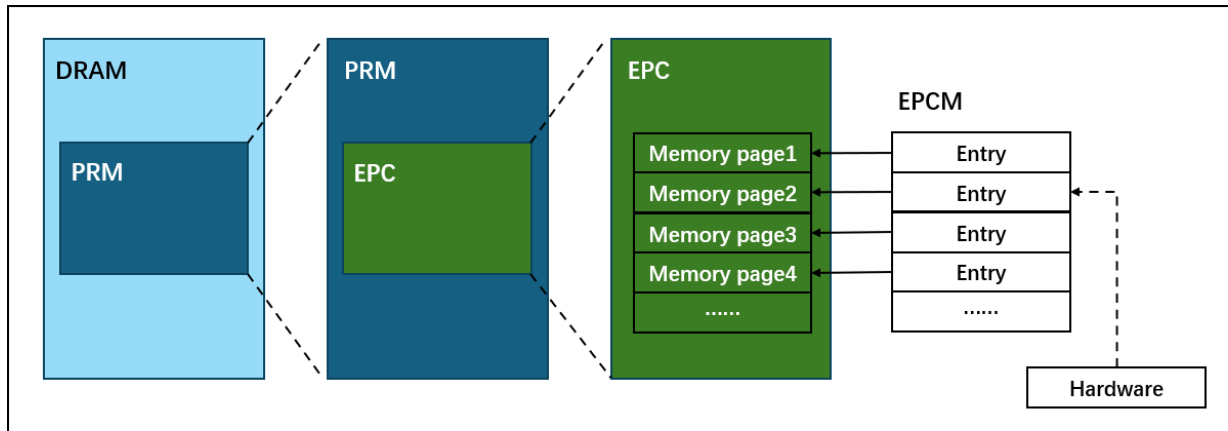


Figure 2. Memory map

When SGX is enabled in the BIOS, the BIOS will reserve a section of memory in DRAM by configuring a set of range registers called PRM. This continuous memory space is at the lowest BIOS level and cannot be accessed by other software including system software. The processor's memory controller safeguards data by blocking peripheral devices from directly accessing PRM memory.

EPC is a collection of 4KB memory allocated by the operating system in PRM to load application data and code segments. Each EPC page can only be assigned to one enclave at a time and only be accessed from inside an enclave. To avoid the untrusted system software allocates the EPC pages to enclaves, the trusted hardware needs to check each EPC entry allocation record. Any unauthorized software requesting access to the EPC page will cause the processor to issue a fault and deny access. The allocation records for EPC entries are stored in the corresponding EPCM entries, maintaining a one-to-one mapping relationship.

The EPCM serves as the gateway to the EPC, storing control information for EPC pages. Similar to a page table in an operating system, it manages key details about EPC pages, such as their usage status, ownership, page types, address mapping, and permission attributes. The EPCM is not directly accessible by software; instead, it is accessed through a hardware module called the Page Miss Handler (PMH). The PMH performs normal accesses to EPC pages by referencing the page table, range registers, and EPCM data, while blocking any unauthorized or abnormal access attempts.

Enabling Intel SGX on ThinkSystem servers

Tip: For details of enabling Intel SGX on ThinkSystem servers, see the Lenovo Press paper [Enabling Intel SGX on Lenovo ThinkSystem Servers](#).

The SGX feature can be enabled by the following steps:

1. In System Setup (F1 at boot), enter the UEFI System Configuration and Boot Management as shown in the following figure:

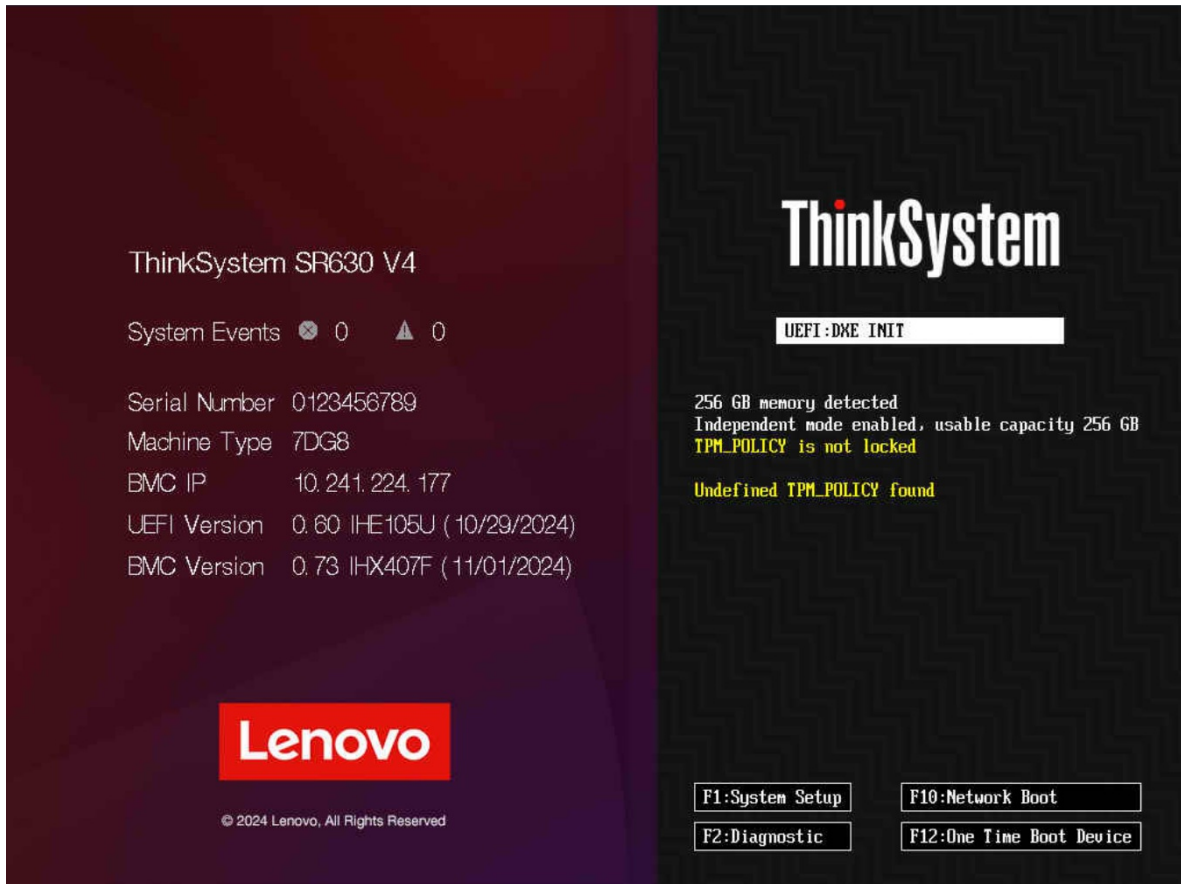


Figure 3. Boot Management in Lenovo System Setup

2. From the BIOS setup menu path, do the following operations, as shown in the following figure.
System Settings → **Processors** → **SW Guard Extension** to enable SGX feature.
System Settings → **Processors** → **SGX PRM Size** to pre-reserve SGX memory.

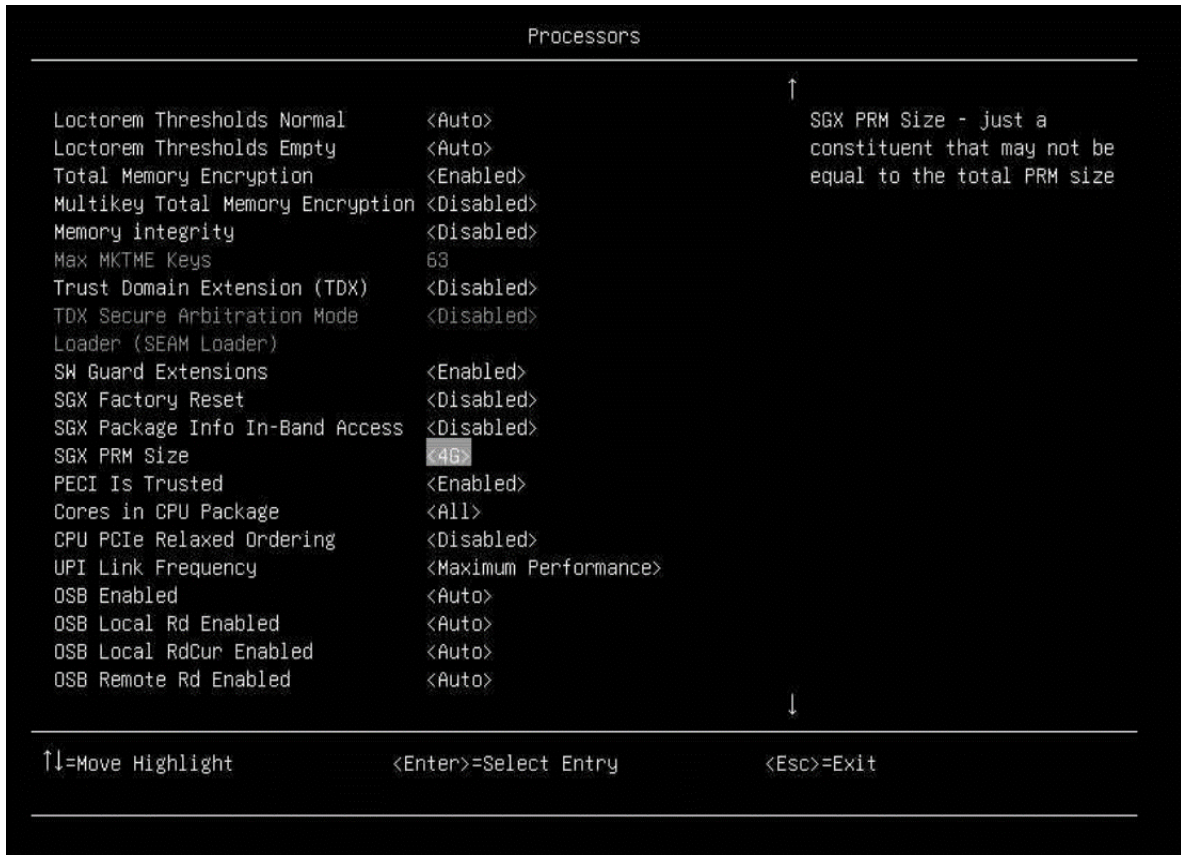


Figure 4. SGX feature in Lenovo System Setup

- For 3rd Gen Scalable processors, there is an extra configuration needs to be set: **System Settings** → **Processors** → **Total Memory Encryption (TME)** to set TME enable.

For detailed UEFI and HW configurations, refer the Lenovo Press paper [Enabling Intel SGX on Lenovo ThinkSystem Servers](#).

Testing and Validating SGX

This section primarily focuses on the OS level, verifying from the software layer whether the hardware properly supports SGX and how to build and use SGX-secured enclaves to test and validate SGX functionalities.

In this section:

- [Hardware support check](#)
- [SGX test suit verification](#)

Hardware support check

This process describes downloading and compiling SGX hardware tools, checking SGX version support, verifying SGX device availability, inspecting kernel-enclave reserved memory, and performing CPUID checks. These checks ensure that the hardware and UEFI fully and correctly support SGX.

- Download SGX Feature Detection Tool.

```
# git clone https://github.com/ayeks/SGX-hardware.git
```

2. Install SGX Feature Detection Tool Dependency Packages.

```
# yum install git make gcc libcap-devel
```

3. Compile SGX Tool.

```
# cd SGX-hardware/ && make
```

4. Make sure SGX device, memory and feature are all in good working order. When SGX is functioning properly, four device nodes should be visible under the /dev/ directory.

```
# ls /dev | grep sgx
```

Four devices are listed below:

```
[root@localhost ~]# ls /dev/ | grep sgx
sgx
sgx_enclave
sgx_provision
sgx_vepc
```

Figure 5. SGX device check

5. Pre-reserved memory: Make sure the operating system reserves memory. A record of the reserved EPC section should appear in the dmesg logs, confirming that the kernel has successfully allocated memory for SGX enclaves.

```
# dmesg | grep sgx
```

```
[root@localhost ~]# dmesg | grep sgx
[ 50.944335] sgx: EPC section 0x1c06000000-0x1ffebfefff
[ 51.054482] sgx: EPC section 0x6280600000-0x62bfffffff
```

Figure 6. SGX EPC section check

6. CPU instruction support: “sgx” and “sgx_lc” flags must be enabled. The two CPU flags indicate that SGX is supported at the hardware level by the CPU, and SGX has been enabled in the UEFI settings.

```
# lscpu | grep sgx
```

```
[root@localhost ~]# lscpu | grep sgx
Flags: fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush dts acpi mmx fxsr sse s
se2 ss ht tm pbe syscall nx pdpe1gb rdtscp lm constant_tsc art arch_perfmon pebs bts rep_good nopl xtopology nonstop_tsc cpuid aperfmperf t
sc_known_freq pni pclmulqdq dtes64 monitor ds_cpl vmx smx est tm2 ssse3 sdbg fma cx16 xtpr pdcm pcid dca sse4_1 sse4_2 x2apic movbe popcnt
tsc_deadline_timer aes xsave avx f16c rdrand lahf_lm abm 3dnowprefetch cpuid_fault epb cat_l3 cat_l2 cdp_l3 cdp_l2 ssbd mba ibrs ibpb stibp
ibrs_enhanced tpr_shadow flexpriority ept vpid ept_ad fsgsbase tsc_adjust sgx bmi1 avx2 smep bmi2 erms invpcid cqm rdt a_avx512f avx512dq
rdseed adx smap avx512ifma clflushopt clwb intel_pt avx512cd sha_ni avx512bw avx512vl xsaveopt xsavec xgetbv1 xsaves cqm_llc cqm_occup_llc
cqm_mbm_total cqm_mbm_local split_lock_detect avx_vnni avx512_bf16 wbnoinvd dtherm ida arat pln pts vnni avx512vbmi umip pku ospke wai1pkg
avx512_vbmi2 gfni vaes vpclmulqdq avx512_vnni avx512_bitalg tme avx512_vppopcntdq la57 rdpid bus_lock_detect cldemote movdiri movdir64b enq
md sgx_lc fsrm md_clear serialize tsxldtrk pconfig arch_lbr ibt amx_bf16 avx512_fp16 amx_tile amx_int8 flush_lid arch_capabilities
```

Figure 7. SGX CPU flags check

7. SGX Detection Tool test. This tool executes the CPUID instruction to read the values of the EAX, EBX, ECX, and EDX registers, extracting SGX version support and hardware capability information from the CPU feature registers. The “Support SGX” entries must be 1 to ensure proper functionality of SGX-related CPU instructions. If the values aren’t “1”, check the hardware compatibility (CPU and memory slots) for SGX support and UEFI configurations.

```
# ./test-sgx
```

```
Supports SGX
SGX Launch Configuration (SGX_LC): 1
SGX Attestation Services (SGX_KEYS): 1
SGX1 leaf instructions (SGX1): 1
SGX2 leaf instructions (SGX2): 1
```

Figure 8. SGX version support check

SGX test suit verification

The SGX test suite requires both SGX-SDK and SGX-PSW. Since the compilation and installation of SGX-PSW depend on the SGX-SDK, the SGX-SDK must be installed first, followed by the SGX-PSW.

1. Download SGX test suit.

```
# git clone https://github.com/intel/linux-sgx.git
```

2. Install test suit dependency.

```
# yum install ocaml ocaml-ocamlbuild wget python3 openssl-devel git cmake
ke perl kernel-devel
# yum groupinstall 'Development Tools'
# yum install openssl-devel libcurl-devel protobuf-devel cmake rpm-build
createrepo yum-utils pkgconf boost-devel protobuf-lite-devel systemd-
libs
```

3. Compile preparation.

```
# cd linux-sgx && make preparation
# cp external/toolset/rhel8.6/* /usr/local/bin/
```

OS needs to find the copied toolset under /usr/local/bin/, you can use the following command to check.

```
# which ar as ld objcopy objdump ranlib
```

4. Compile SGX-SDK.

```
# make sdk && make sdk_install_pkg
```

5. Install SGX-SDK.

```
# cd linux/installer/bin/ && ./sgx_linux_x64_sdk_[version].bin
```

sgx_linux_x64_sdk_[version].bin will be generated under the path of linux/installer/bin/ after compilation.

When asking the installation path, you can input the path such as "/opt/intel/".

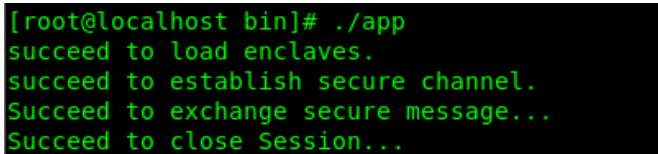
6. Source environment.

```
# source /opt/intel/sgxsdk/environment
```

7. Verify SGX-SDK.

```
# cd /opt/intel/sgxsdk/SampleCode/LocalAttestation/  
# make SGX_MODE=SIM  
# cd bin && ./app
```

If SDK is in good working order, it will show the following message:



```
[root@localhost bin]# ./app  
succeed to load enclaves.  
succeed to establish secure channel.  
Succeed to exchange secure message...  
Succeed to close Session...
```

Figure 9. Successful running of SGX-SDK

8. Compile SGX-PSW.

```
# cd /path/to/linux-sgx && make psw && make psw_install_pkg  
# cd psw/ae/le && make
```

9. Install SGX-PSW.

```
# cd linux/installer/bin/ && ./sgx_linux_x64_psw_[version].bin
```

sgx_linux_x64_psw_[version].bin will be generated under the path of linux/installer/bin/ after compilation.

10. Verify SGX-PSW.

```
# cd /opt/intel/sgxsdk/SampleCode/LocalAttestation/  
# make  
# cd bin && ./app
```

If SDK is in good working order, it will show the following message:

```
[root@localhost bin]# ./app  
succeed to load enclaves.  
succeed to establish secure channel.  
Succeed to exchange secure message...  
Succeed to close Session...
```

Figure 10. Successful running of SGX-PSW

Support Scope

This section lists the kernel and OS distributions/versions that support Intel SGX.

The following kernel versions support Intel SGX:

- Kernel version 5.11 or later.

Supported Intel platforms:

- 3rd Gen Intel Xeon Scalable processors or later

The following 64-bit operating systems support Intel SGX:

- Ubuntu 20.04 LTS Desktop, or later
- Ubuntu 20.04 LTS Server, or later
- Ubuntu 22.04 LTS Server, or later
- Ubuntu 23.10 Server, or later
- Red Hat Enterprise Linux Server 9.2, or later
- CentOS Stream 9, or later
- CentOS 8.3, or later
- SUSE Linux Enterprise Server 15.4, or later
- Anolis OS 8.6, or later
- Debian 10, or later

Troubleshooting

This section lists common issues you might encounter during your testing, and recommended methods to resolve the problems.

1. No “sgx” on lscpu flags

This indicates that SGX has not been successfully enabled; perform the following checks:

- Whether CPU platform supports SGX
- Whether the location and quantity of physical memory modules meet the requirements
- Whether UEFI configuration is correct

2. SGX-hardware test shows SGX1 leaf or SGX2 leaf is 0

This indicates that SGX is supported on this platform but not enabled correctly.

You will get the following test results if you execute “./test_sgx”:

```
SGX Launch Configuration (SGX_LC): 1
SGX Attestation Services (SGX_KEYS): 0
SGX1 leaf instructions (SGX1): 0
SGX2 leaf instructions (SGX2): 0
```

If execute “dmesg | grep sgx”, you will also get the following

```
There are zero EPC sections.
```

Verify if there are any issues with the CPU hardware. If necessary, consider replacing the CPU.

3. Compile Errors during building SGX test suit

Perform the following check:

- Whether all dependent packages are installed.
- Whether the network can connect to Github, the SGX test suit needs additional dependent packages to be pulled from GitHub for compilation.
- Whether c++ std version meets the SGX test suit requirements (The default requirement is c++ std 11, but on some newer OS versions, the default c++ std is 14. Modify the target directory CMakeLists.txt as "set(CMAKE_CXX_STANDARD 14)" to solve this issue).

4. “./app” shows “failed to load enclave”

If the kernel version is greater than 5.11 and the hardware platform supports SGX, execute “./app” still report “failed to load enclave”, then try referencing this link (https://download.01.org/intel-sgx/latest/linux-latest/docs/Intel_SGX_SW_Installation_Guide_for_Linux.pdf) and using the pre-build version to retry. This failure might be caused by SGX test suit version or dependencies.

References

See the following web pages for more information:

- General overview of Intel SGX
<https://sys.cs.fau.de/extern/lehre/ws22/akss/material/intel-sgx.pdf>
- Intel SGX for Linux on github
<https://github.com/intel/linux-sgx?tab=readme-ov-file>
- What is Intel SGX and What are the Benefits?
<https://phoenixnap.com/kb/intel-sgx>
- Intel SGX - Reduce the Attack Surface Around Your Data to Unlock New Opportunities
<https://www.intel.com/content/www/us/en/products/docs/accelerator-engines/software-guard-extensions.html>
- Wikipedia - Intel SGX
https://en.wikipedia.org/wiki/Software_Guard_Extensions
- Enabling Intel SGX on Lenovo ThinkSystem Servers
<https://lenovopress.lenovo.com/lp1471-enabling-intel-sgx-on-lenovo-thinksystem-servers>
- Intel SGX Software Installation Guide For Linux OS
https://download.01.org/intel-sgx/latest/linux-latest/docs/Intel_SGX_SW_Installation_Guide_for_Linux.pdf

Author

Dong Wang is a Linux engineer in the Lenovo Infrastructure Solution Group in Beijing, China.

Thanks to the following people for their contributions to this project:

- Adrian Huang

Related product families

Product families related to this document are the following:

- [Processors](#)

Notices

Lenovo may not offer the products, services, or features discussed in this document in all countries. Consult your local Lenovo representative for information on the products and services currently available in your area. Any reference to a Lenovo product, program, or service is not intended to state or imply that only that Lenovo product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any Lenovo intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any other product, program, or service. Lenovo may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

Lenovo (United States), Inc.
8001 Development Drive
Morrisville, NC 27560
U.S.A.
Attention: Lenovo Director of Licensing

LENOVO PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. Lenovo may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

The products described in this document are not intended for use in implantation or other life support applications where malfunction may result in injury or death to persons. The information contained in this document does not affect or change Lenovo product specifications or warranties. Nothing in this document shall operate as an express or implied license or indemnity under the intellectual property rights of Lenovo or third parties. All information contained in this document was obtained in specific environments and is presented as an illustration. The result obtained in other operating environments may vary. Lenovo may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any references in this publication to non-Lenovo Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this Lenovo product, and use of those Web sites is at your own risk. Any performance data contained herein was determined in a controlled environment. Therefore, the result obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

© Copyright Lenovo 2025. All rights reserved.

This document, LP2183, was created or updated on March 20, 2025.

Send us your comments in one of the following ways:

- Use the online Contact us review form found at:
<https://lenovopress.lenovo.com/LP2183>
- Send your comments in an e-mail to:
comments@lenovopress.com

This document is available online at <https://lenovopress.lenovo.com/LP2183>.

Trademarks

Lenovo and the Lenovo logo are trademarks or registered trademarks of Lenovo in the United States, other countries, or both. A current list of Lenovo trademarks is available on the Web at <https://www.lenovo.com/us/en/legal/copytrade/>.

The following terms are trademarks of Lenovo in the United States, other countries, or both:

Lenovo®

ThinkSystem®

The following terms are trademarks of other companies:

Intel® and Xeon® are trademarks of Intel Corporation or its subsidiaries.

Linux® is the trademark of Linus Torvalds in the U.S. and other countries.

Other company, product, or service names may be trademarks or service marks of others.