



# Intel® Software Guard Extensions SSL (Intel® SGX SSL) Library

**Linux Developer Guide**

---

# 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.

Table of Contents

Legal Information .....2

1. Package Content.....4

2. Using Intel® SGX SSL Library .....5

3. Supported APIs.....6

4. Appendix A: Supported APIs .....9

# 1. Package Content

---

Intel® SGX SSL library is released as a component of the Intel® Software Guard Extensions (Intel® SGX) SDK. Private release package can be provided by request for evaluation purposes.

The release package contains relevant include files (both header and edl files), libraries and relevant documentation.

The following table lists the libraries provided in the release package:

Library Name	Description
<code>libsgx_tsgxssl_crypto.a</code>	Intel® SGX SSL* cryptographic library, built based on OpenSSL 1.1.0 crypto library
<code>libsgx_tsgxssl.a</code>	Trusted library, providing implementation for missing system APIs required by Intel® SGX SSL cryptographic library
<code>libsgx_usgxssl.a</code>	Untrusted library, providing implementation for system calls outside an enclave required to resolve external dependencies of Intel® SGX SSL* cryptographic and TLS libraries.

All the libraries are built for Linux\* configurations.

Intel® SGX SSL\* cryptographic library is OpenSSL libraries built with a few changes needed to work inside an enclave.

## 2. Using Intel® Software Guard Extensions SSL Library

---

If you already have a basic application and an enclave project, to use the Intel® SGX SSL library in an Intel® Software Guard Extensions (Intel® SGX) application project, follow the listed steps:

- Use following steps to set up generating proper interface between trusted and untrusted components
  1. In your EDL file add:
 

```
from "sgx_tsgxssl.edl" import *;
```
  2. To the `sgx_edger8r` command running on your enclave EDL file for generating either trusted or untrusted proxy and bridge routines, add the path to the `sgx_tsgxssl.edl` with the `--search path` option
- In the **Enclave** project, use the following steps to set up the environment for the Intel® SGX SSL
  1. Use `-L` flag to provide the linker with the path to the trusted Intel® SGX SSL libraries `libsgx_tsgxssl_crypto.a` and `libsgx_tsgxssl.a`, with `-L$(SGXSSL_TRUSTED_LIB_PATH)`
  2. Use `-Wl,--whole-archive -lsgx_tsgxssl -Wl,--no-whole-archive -lsgx_tsgxssl_crypto -lsgx_tsetjmp` to provide the linker with the names of Intel® SGX SSL trusted libraries and the `setjmp` library which is also needed (comes with Intel® SGX SDK)
 

**NOTE:** `-lsgx_tsetjmp` is only required when using old Intel® SGX SDK version, 1.9 or lower.
  3. Use `-I` compilation flag to specify the path to the Intel® SGX SSL header files, like `-I$(SGXSSL_INCLUDE_PATH)`
  4. The Intel® SGX SSL include path also includes a reduced “pthread.h” file which only have 3 definitions, it is included from `openssl/crypto.h`. Make sure it is not in the path of your regular application as it may cause compilation errors
  5. Include `tsgxsslio.h` file to avoid error on undeclared `FILE` symbol. You can do it either directly from your source files, or by using `-include "tsgxsslio.h"` compiler flag
  6. Initialize OpenSSL crypto library by calling `OPENSSL_init_crypto(0, NULL)`. Make sure to initialize before the first call to OpenSSL APIs.
- In the **Application** project, use the following steps to set up the environment for the Intel® SGX SSL library:
  1. Use `-L` flag to provide the linker with the path to the untrusted Intel® SGX SSL library `libsgx_usgxssl.a`, with `-L$(SGXSSL_UNTRUSTED_LIB_PATH)`
  2. Use `-lsgx_usgxssl` to provide the linker with the names of Intel® SGX SSL untrusted libraries

**NOTE:** In the current Intel® SGX SDK, the `release` mode does not generate the `enclave.signed.so`, but rather prepare a signing material because it should be signed in a secure machine that protects the private key. Enclaves signed with single-step signing method using ISV's test private key can only be launched in `debug` or `prerelease` modes.

## 3. Supported APIs

The Intel® SGX SSL Library exposes two different set of APIs:

- Supported OpenSSL APIs - representing a subset of the OpenSSL APIs supported by the Intel® SGX SSL library. They are fully compliant with unmodified OpenSSL APIs. Other APIs are neither validated, not filtered out. All supported OpenSSL APIs are listed in [Appendix A](#).
- Manageability APIs are exposed by our trusted library to provide following services:

API	Description
<code>SGXSSLSetPrintToStdoutStderrCB</code>	Set callback function to intercept printouts sent by Intel® SGX SSL cryptographic and TLS libraries to <code>stdout/stderr</code> . If not used, the printouts will be silently omitted.
<code>SGXSSLGetSgxSSLVersion</code>	Get the Intel® SGX SSL library version.
<code>SGXSSLSetUnreachableCodePolicy</code>	Set unreachable code policy. Unreachable code consists of functions and flows that under our implementation should never be reached. That is why, by default, reaching unreachable code will cause an enclave to be aborted.

### SGXSSLSetPrintToStdoutStderrCB

The `SGXSSLSetPrintToStdoutStderrCB` function sets callback function to intercept Intel® SGX SSL cryptographic and TLS libraries printouts sent to `stdout/stderr`. If not used, the printouts will be silently omitted.

#### Syntax

```
void SGXSSLSetPrintToStdoutStderrCB(
    PRINT_TO_STDOUT_STDERR_CB cb
);
```

#### Parameters

##### cb [in]

Callback function to intercept OpenSSL printouts to `stdout/stderr`.

#### Return value

This function does not return a value.

#### Description

The `SGXSSLSetPrintToStdoutStderrCB` function registers a callback function to intercept Intel® SGX SSL cryptographic and TLS printouts sent to `stdout/stderr`.

If not used, the printouts will be silently omitted.

#### Requirements

Header	<code>tSgxSSL_api.h</code>
Library	<code>sgx_tsgxssl.lib</code>

## SGXSSLGetSgxSSLVersion

The `SGXSSLGetSgxSSLVersion` function returns the Intel® SGX SSL libraries version.

### Syntax

```
const char* SGXSSLGetSgxSSLVersion(
    void
);
```

### Parameters

None

### Return value

This function returns the Intel® SGX SSL libraries version string.

### Description

The `SGXSSLGetSgxSSLVersion` function returns the Intel® SGX SSL libraries version string.

### Requirements

Header	<code>tsgxssl_api.h</code>
Library	<code>sgx_tsgxssl.lib</code>

## SGXSSLSetUnreachableCodePolicy

The `SGXSSLSetUnreachableCodePolicy` function sets unreachable code policy.

If not used, reaching unreachable code will cause an enclave to be aborted.

### Syntax

```
void SGXSSLSetUnreachableCodePolicy(
    UnreachableCodePolicy_t policy
);
```

### Parameters

**policy [in]**

The valid value is `UNREACH_CODE_ABORT_ENCLAVE` or `UNREACH_CODE_REPORT_ERR_AND_CONTINUE`.

- `UNREACH_CODE_ABORT_ENCLAVE` value means that reaching unreachable code will cause an enclave to be aborted. This is the default policy, applied by Intel® SGX SSL library.
- `UNREACH_CODE_REPORT_ERR_AND_CONTINUE` value means that reaching unreachable code will cause reporting an error through return value and/or setting last `error/errno`.

### Return value

None.

### Description

The `SGXSSLSetUnreachableCodePolicy` function sets unreachable code policy. Unreachable code consists of functions and flows that under our implementation should never be reached. Reaching them may indicate that severe error/memory corruption happened. That is why, by default, reaching unreachable code will cause an enclave to be aborted.

For customers, which in any case prefer to continue execution, additional mode, reporting an error through return value and/or setting last `error/errno`, is supported.

#### Requirements

Header	<code>tSgxSSL_api.h</code>
Library	<code>sgx_tsgxssl.lib</code>



## 4. Appendix A: Supported APIs

Intel® SGX SSL library supports the following APIs:

Purpose	Type	OpenSSL APIs
Digest	MD5 SHA-1 SHA-2 (224, 256, 384, 512)	EVP_MD_CTX_new EVP_MD_CTX_free EVP_DigestInit_ex EVP_DigestUpdate EVP_DigestFinal_ex EVP_md5 EVP_sha1 EVP_sha224, EVP_sha256, EVP_sha384, EVP_sha512
Keyed Hash	HMAC	HMAC_CTX_init HMAC_CTX_cleanup HMAC_Init_ex HMAC_Update HMAC_Final
Public Key Cryptography	RSA 1024, 2048, 4096 ECDSA NIST P-256, P-384, P-521 ECDH NIST P-256, P-384, P-521	EC_KEY_new_by_curve_name EC_KEY_set_asn1_flag EC_KEY_generate_key EC_KEY_free RSA_new RSA_free RSA_generate_key_ex RSA_private_decrypt EVP_PKEY_new EVP_PKEY_assign_EC_KEY EVP_PKEY_assign_RSA EVP_PKEY_free EVP_MD_CTX_create EVP_MD_CTX_destroy EVP_SignInit_ex EVP_SignUpdate EVP_SignFinal EVP_VerifyInit_ex EVP_VerifyUpdate EVP_VerifyFinal

Symmetric Encryption	AES-GCM 128, 256	EVP_CIPHER_CTX_init EVP_CIPHER_CTX_ctrl EVP_CIPHER_CTX_cleanup EVP_CipherInit_ex EVP_CipherUpdate EVP_CipherFinal_ex EVP_aes_128_gcm EVP_aes_256_gcm
Other	Public key cryptography: RSA, EC	BN_new BN_set_word OBJ_txt2nid i2d_PublicKey I2d_PrivateKey RAND_add RAND_seed