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

Windows 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.

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

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

Table of Contents

1. Package Content

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
`libsgx_tsgxssl_crypto.a`	Intel® SGX SSL* cryptographic library, built based on OpenSSL 1.1.1d crypto library
`libsgx_tsgxssl.a`	Trusted library, providing implementation for missing system APIs required by Intel® SGX SSL cryptographic library
`libsgx_usgxssl.a`	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 Windows Win32 and X64 configuration. And the libraries with CVE-2020-0551 Mitigation enabled, 2 levels, X64 release only, are also available at the corresponding installation paths.

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

In visual studio you will need to do the following (assuming you already have basic App + Enclave project):

- As a start, you may extract Intel® SGX SSL package to solution's directory. (You may also extract it into Intel® SGX SDK directory, or any other location, as long as you refer to the right location in projects' settings)

- In your EDL file add:
from "sgx_tsgxssl.edl" import *;

- Before using any OpenSSL API, you must include OpenSSL header that declares it. (e.g. <openssl\crypto.h>, <openssl\sha.h>, <openssl\ripemd.h>)

- Add "#include <windows.h>" before the "#include <openssl\xxx>" statements. The "windows.h" file is found in the Intel® SGX SSL include directory, and contains several definitions required by many of the OpenSSL headers.

- In the **Enclave** project (do these steps to all of your build environments):

- Select **Properties->Linker->Input->Additional Dependencies**:
    Add "`libsgx_tsgxssl_crypto.lib; libsgx_tsgxssl.lib;`"

- Select **Properties->Linker->General->Additional Library Directories**:
    Add the folder where you placed the libraries. You'd better use the built-in macros like $(SolutionDir)$(Platform)\$(Configuration)\ etc., so you can control the different builds. To take the CVE-2020-0551 Mitigation enabled library, you can create either of the configuration based on the release configuration for your solution: CVE-2020-0551-Load-Release or CVE-2020-0551-CF-Release.

- To add the folder where you placed the EDL file, right click your EDL file, then select **Properties->Custom Build Tool->Command Line**:
    Add the EDL file path to the '`--search path`' separated with ';'

- Select **Properties**->**C/C++**->**General**->**Additional Include Directories** and add the folder where Intel® SGX SSL header files are located. (`<path to the package>\include`)

- In the **Application** project, use the following steps to set up the environment for the Intel® SGX SSL library:

- Select **Properties->Linker->Input->Additional Dependencies**:
    Add ''`libsgx_usgxssl.lib;Ws2_32.lib`''

- Select **Properties->Linker->General->Additional Library Directories**:

Add the folder where you placed the libraries (you better use the built in macros like $(SolutionDir)$(Platform)\$(Configuration)\ etc. so you can control the different builds)

- To add the folder where you placed the EDL file, right click your EDL file, then select **Properties->Custom Build Tool->Command Line**:
    Add the EDL file path to the '--search path' separated with ';'

- If your project does not use Intel compiler, add the path to the Intel compiler libraries through **Properties->Linker->General->Additional Library Directories**

- Please note that in the current Intel® SGX SDK, the 'release' mode does not generate the enclave.signed.dll but rather prepare a signing material (since it should be signed in a secure machine that protects the private key etc.). 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
`SGXSSLSetPrintToStdoutStderrCB`	Set callback function to intercept printouts sent by Intel® SGX SSL cryptographic and TLS libraries to `stdout/stderr`. If not used, the printouts will be silently omitted.
`SGXSSLGetSgxSSLVersion`	Get the Intel® SGX SSL library version.
`SGXSSLSetUnreachableCodePolicy`	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	tSgxSSL_api.h
Library	sgx_tsgxssl.lib

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	tSgxSSL_api.h
Library	sgx_tsgxssl.lib

SGXSSLSetUnreachableCodePolicy

The `SGXSSLSetUnreachableCodePolicy` function sets unreachable code policy.

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

Syntax

```
void SGXSSLSetUnreachableCopdePolicy(
UnreachableCopdePolicy_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_CONTNUE` 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	`tSgxSSL_api.h`
Library	`sgx_tsgxssl.lib`

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