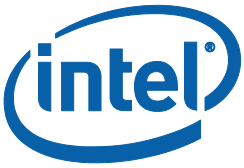

Intel® SgxSSL Library

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.

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.

Revision History

Revision Number	Description	Revision Date
1.0	First release, with OpenSSL 1.1.0b and SGX SDK 1.6 for Linux	September 2016
1.1	Updated to OpenSSL 1.1.0c, SGX SDK 1.7 for Linux	November 2016
1.2	Updated to OpenSSL 1.1.0e	February 2017

Table of Contents

1. Introduction

1.1. Intel® SgxSSL Library

The Intel® SgxSSL cryptographic library is intended to provide cryptographic services for Intel® Software Guard Extensions (SGX) enclave applications.

The Intel® SgxSSL cryptographic library is based on the underlying OpenSSL* Open Source project, providing a full-strength general purpose cryptography library.

The API exposed by the Intel® SgxSSL library is fully compliant with unmodified OpenSSL APIs.

NOTE: Only a specific subset of APIs available in OpenSSL is supported by the Intel® SgxSSL cryptographic library. Unsupported OpenSSL APIs included in the Intel® SgxSSL cryptographic library are not validated or recommended. See Appendix A for the supported OpenSSL APIs.

In addition, the Intel® SgxSSL library exposes a closed set of manageability APIs, a list of which is provided in Supported APIs.

1.2. Terminology

Term	Description
SGX	Software Guard Extension
EDL	Enclave Definition Language. EDL files define the enclave interface for trusted-to-untrusted and untrusted-to-trusted transitions.

1.3. Legal Considerations

The Intel® SgxSSL Library is based on the OpenSSL* open source libraries licensed under a dual license, i.e. both the conditions of the OpenSSL license and the original SSLeay license apply.

More information regarding the OpenSSL license is available at https://www.openssl.org/source/license.html

You MUST be aware of license requirements and/or limitations of the underlying OpenSSL library and fully conform to it.

NOTE: This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (https://www.openssl.org)

1.4. Architecture Overview

The Intel® SgxSSL library consists of the following components:

- Intel® SgxSSL cryptographic library representing OpenSSL* 1.1.0 cryptographic library built to run inside an enclave.
- A trusted library providing implementation for missing system APIs inside an enclave.
- An untrusted library providing implementation of missing system APIs outside an enclave.

The following figure shows how Intel(R) SgxSSL library is used in an SGX application.

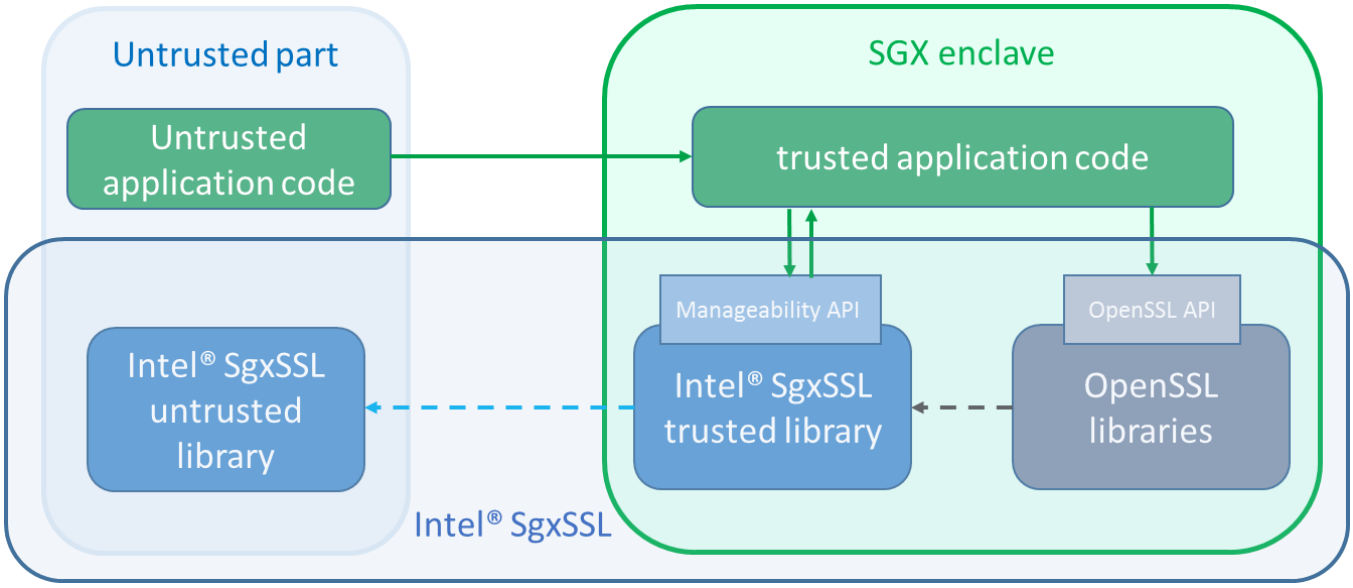

Figure 1 Intel(R) SgxSSL library Usage

Here is the flow of execution as illustrated in Figure 1:

1. The user's untrusted application code calls the trusted code with a function declared in the EDL file
2. The user's trusted code may use manageability API for different purposes, like to get Intel® SgxSSL library version, to register a callback function to intercept the Intel® SgxSSL libraries printout messages, and so on.
3. The user's trusted code continues execution and at a certain point calls an API supported by the Intel® SgxSSL cryptographic or TLS libraries. The supported API is a subset of the unmodified OpenSSL API.
4. The call is passed to the Intel® SgxSSL cryptographic library. Some functions are internal and do not rely on system APIs (for example, SHA256), so the functions complete and return.
5. Other functions require some system APIs, so the execution passes to the Intel® SgxSSL trusted library code that implements them. If the system API can be implemented internally (for example, pthread_once), it returns after completion without leaving the enclave.
6. Other APIs must leave the trusted code and are executed in the untrusted area (for example, ftime)

On an S3/S4 power event the internal state of the operation (for instance, during BASE64 encode/decode API usage) executed by the Intel(R) SGX SgxSSL library will be lost as part of the entire enclave loss. The Intel® SgxSSL library does not manage saving or restoring the state on suspend or resume operations. It is the customer's application responsibility to save an internal Intel® SgxSSL state on suspend and to restore it on resume when applicable.

1.5. Security Recommendations.

Intel® SgxSSL provides support for OpenSSL* inside an enclave. Security assets, like cryptographic keys, client certificate private keys, and plain data (both network traffic and cryptographic payload) do not need to leave an enclave. Thus they are protected by the Intel® SGX technology. Intel® SgxSSL library provides integrity and confidentiality of security assets and protects them from both malicious software and a simple hardware attack.

Intel® SgxSSL library relies on an implementation of OpenSSL and Intel® SGX to handle side channel attacks. Our architecture is designed to not leak additional information through the OCALLs, but it doesn't protect against side channel attacks. So, in case of side channel attacks it is as secure OpenSSL without Intel® SGX.

Getting current system time is not supported inside an enclave and is therefore implemented by Intel® SgxSSL library as OCALL. This approach allows an attacker to manipulate the time values coming from an untrusted component. Time values are used by Intel® SgxSSL library for time related certificate verification checks as well as for TLS session expiration checks. To reduce the risk of a time attack on an enclave application that uses the Intel(R) SgxSSL, we recommend you use server certificate pinning.

An enclave application, built with Intel® SgxSSL library, is responsible for preserving the protection features of Intel® SgxSSL library. Follow the listed expectations of and recommendations for the customer enclave application:

- The customer's application is responsible to build production enclave as non-debug enclave.
- The customer's application should utilize Intel® SGX architecture and Intel® SGX software to protect security assets. For instance, the customer's application should use certificate pinning. Server certificate should be provisioned into an enclave and protected by enclave sealing capabilities.
- The customer's application should not expose security assets by trusted to untrusted transitions
- The customer's application should sanitize input data coming from untrusted components
- The customer's application should configure Intel® SgxSSL not to support obsolete protocols and cryptographic suites
- The customer's application should use only the OpenSSL APIs that are supported by the Intel® SgxSSL library.
- The customer's application should use OpenSSL APIs correctly to verify server certificate.

2. Release Information

2.1. Package Content

Intel® SgxSSL library is released as a component of the Intel® Software Guard Extensions (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
`libsgx_tsgxssl_crypto.a`	Intel® SgxSSL* cryptographic library, built based on OpenSSL 1.1.0 crypto library
`libsgx_tsgxssl.a`	Trusted library, providing implementation for missing system APIs required by Intel® SgxSSL cryptographic library
`libsgx_usgxssl.a`	Untrusted library, providing implementation for system calls outside an enclave required to resolve external dependencies of Intel® SgxSSL* cryptographic and TLS libraries.

All the libraries are built for Linux* configurations.

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

2.2. Using Intel® SgxSSL Library

If you already have a basic application and an enclave project, to use the Intel® SgxSSL library in an 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® SgxSSL

  1. Use `-L` flag to provide the linker with the path to the trusted SgxSSL 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 SgxSSL trusted libraries and the setjmp library which is also needed (comes with SGX SDK)

3. Use `-I` compilation flag to specify the path to the SgxSSL header files, like `-I$(SGXSSL_INCLUDE_PATH)`
4. The SGXSSL 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

- In the **Application** project, use the following steps to set up the environment for the Intel(R) SgxSSL library:

1. Use `-L` flag to provide the linker with the path to the untrusted SgxSSL library `libsgx_usgxssl.a`, with `-L$(SGXSSL_UNTRUSTED_LIB_PATH)`
2. Use `-lsgx_usgxssl` to provide the linker with the names of SgxSSL 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.

2.3. Supported APIs

The Intel® SgxSSL Library exposes two different set of APIs:

- Supported OpenSSL APIs - representing a subset of the OpenSSL APIs supported by the Intel® SgxSSL 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
`setPrintToStdoutStderrCB`	Set callback function to intercept printouts sent by Intel® SgxSSL cryptographic and TLS libraries to `stdout/stderr`. If not used, the printouts will be silently omitted.
`getSgxSSLVersion`	Get the Intel® SgxSSL library version.
`setUnreachableCodePolicy`	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.

setPrintToStdoutStderrCB

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

Syntax

```
void setPrintToStdoutStderrCB(
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 `setPrintToStdoutStderrCB` function registers a callback function to intercept Intel® SgxSSL 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`

getSgxSSLVersion

The `getSgxSSLVersion` function returns the Intel® SgxSSL libraries version.

Syntax

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

Parameters

None

Return value

This function returns the Intel® SgxSSL libraries version string.

Description

The getSgxSSLVersion function returns the Intel® SgxSSL libraries version string.

Requirements

Header	`tSgxSSL_api.h`
Library	`sgx_tsgxssl.lib`

setUnreachableCodePolicy

The `setUnreachableCodePolicy` function sets unreachable code policy.

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

Syntax

```
void setUnreachableCopdePolicy(
```

```
    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® SgxSSL 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 `setUnreachableCodePolicy` 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`

3. Appendix A: Supported APIs

Intel® SgxSSL 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