AOCL Crypto OpenSSL Provider Documentation

OpenSSL Plugin Documentation

Contents

List of Figures

Part I

Key Terminologies

- SSL- Secure Socket Layer
- SSH - Secure Shell Host
- Cipher Suite - A set of algorithms which can be used to exchange keys, verify integrity, and provide authenticity.
- Kernel - Piece of code which performs/executes all the core tasks and takes most amount of time in the library.
- Plugin - A foreign piece of code which can be used to extend a program/library.
- Provider - Another name for plugin used by OpenSSL.
- Benchmarking - Performance analysis of a program.

Part II

Introduction

Chapter 1

About OpenSSL

OpenSSL is an opensource SSL library which supports various encryption decryption standards as well as cipher suites. OpenSSL is used in well known programs such as Nginx, SSH etc. Most of the OpenSSL kernels have a hardware optimized version inside it. Sometimes users of OpenSSL might reqire more optimized versions of the kernels, for this particular purpose OpenSSL has a plugin infrastructure which will allow anyone to write plugins called as providers. This is the interface which is used by ALCP to communicate with the OpenSSL infrastructure.

1.1 Using OpenSSL-Compat Lib

ALCP support for OpenSSL is provided by a provider. Provider lib name is `libopenssl-compat.so`. Please configure openssl to use provider by default or setup provider loading inside application itself.

To bench with the provider, use the following example assuming you are executing command from the root of the package directory.

```
    openssl speed -provider-path $PWD/lib  -provider libopenssl-compat -evp
aes-128-gcm
```

Part III

Using provider in a C program

Instructions to use provider in a C program is given in this link

```c
#include <stdio.h>
#include <stdlib.h>

#include <openssl/provider.h>

int main(void)
{
    OSSL_PROVIDER *alcp_provider;

    OSSL_PROVIDER_set_default_search_path("/path/to/alcp/lib")

    alcp_provider = OSSL_PROVIDER_load(NULL, "libopenssl-compat");
    if (NULL == alcp_provider) {
        printf("Failed to load ALCP provider\n");
        exit(EXIT_FAILURE);
    }

    /* Rest of application */

    OSSL_PROVIDER_unload(alcp_provider);
    exit(EXIT_SUCCESS);
}
```

OSSL_PROVIDER_set_default_search_path - Where OpenSSL searches for the provider binary.

OSSL_PROVIDER_load - Name of the provider to load.

OSSL_PROVIDER_unload - Unload the named provider. # Configuring OpenSSL

1.2 Configuring provider to be loaded by default.

For more information please take a look at this. Modify or replace openssl.cnf with this.

```
openssl_conf = openssl_init

[openssl_init]
providers = provider_sect

[provider_sect]
alcp  = alcp_sect
default = default_sect
base = base_sect

[default_sect]
```

```
activate = 1

[alcp_sect]
module = /path/to/libopenssl-compat.so
activate = 1

[base_sect]
activate = 1
```

Above configuration will allow you to offload functionalities if supported by AOCL-Cryptography. OpenSSL will still dispatch to it's own implementation for ones we have not implemented yet. To find out where openssl looks for `openssl.cnf`, type the command `openssl info -configdir`.

1.3 Code taking advantage of configuration

```c
#include <stdio.h>
#include <stdlib.h>

#include <openssl/provider.h>

int main(void)
{
    OSSL_PROVIDER *alcp_provider;

    // Will find the openssl provider shared object from configuration
    alcp_provider = OSSL_PROVIDER_load(NULL, "alcp");

    if (NULL == alcp_provider) {
        printf("Failed to load ALCP provider\n");
        exit(EXIT_FAILURE);
    }

    /* Rest of application */

    OSSL_PROVIDER_unload(alcp_provider);
    exit(EXIT_SUCCESS);
}
```

Part IV

Benchmarking

Chapter 2

OpenSSL Speed

OpenSSL has an inbuilt benchmarking mechanism called `speed`. When you invoke openssl speed using command line (`openssl speed`), openssl benchmarks all the supported algorithms. After the benchmark you will be given a summary, in the summary it will say how much bytes was encrypted/decrypted in a second.

2.1 OpenSSL Speed Arguments

Important Arguments

1) -evp - Use EVP API.
2) -decrypt - Do a decrypt benchmark.
3) -seconds - Edit the benchmark time default 3 seconds.
4) -provider - File name of provider without extension.
5) -provider-path - Path where to look for provider binary file.
6) algorithm - Which algorithm to benchmark.

example -

1. EVP API is used with cipher AES mode CBC with 128bit keysize in encrypt setting.

```
openssl speed -evp aes-128-cbc
```

Chapter 3

Benchmarking with OpenSSL Speed.

In this section we will discuss how to use provider without configuration files. OpenSSL binary can be given arguments to load provider from a custom directory. If openssl.cnf is already configured to use the custom directory and the provider, then specifying provider in argument is not necessary.

To load the OpenSSL provider for AOCL Crypto, we have to provide the path of the provider and the name of the .so file.

```
openssl speed -provider-path /path/to/alcp/lib -provider libopenssl-compat
```

Now we can combine this command with EVP API and provide an algorithm to run the encrypt benchmark

```
openssl speed -provider-path /path/to/alcp/lib -provider libopenssl-compat \
-evp aes-128-cbc
```

3.1 All supported possible commands for OpenSSL speed

3.1.1 AES-CBC

```
openssl speed -provider-path /path/to/alcp/lib -provider libopenssl-compat \
-evp aes-128-cbc

openssl speed -provider-path /path/to/alcp/lib -provider libopenssl-compat \
-evp aes-192-cbc

openssl speed -provider-path /path/to/alcp/lib -provider libopenssl-compat \
-evp aes-256-cbc
```

3.1.2 AES-CTR

```
openssl speed -provider-path /path/to/alcp/lib -provider libopenssl-compat \
-evp aes-128-ctr

openssl speed -provider-path /path/to/alcp/lib -provider libopenssl-compat \
-evp aes-192-ctr

openssl speed -provider-path /path/to/alcp/lib -provider libopenssl-compat \
-evp aes-256-ctr
```

3.1.3 AES-CFB

```
openssl speed -provider-path /path/to/alcp/lib -provider libopenssl-compat \
-evp aes-128-cfb

openssl speed -provider-path /path/to/alcp/lib -provider libopenssl-compat \
-evp aes-192-cfb

openssl speed -provider-path /path/to/alcp/lib -provider libopenssl-compat \
-evp aes-256-cfb
```

3.1.4 AES-OFB

```
openssl speed -provider-path /path/to/alcp/lib -provider libopenssl-compat \
-evp aes-128-ofb

openssl speed -provider-path /path/to/alcp/lib -provider libopenssl-compat \
-evp aes-192-ofb

openssl speed -provider-path /path/to/alcp/lib -provider libopenssl-compat \
-evp aes-256-ofb
```

3.1.5 AES-GCM

```
openssl speed -provider-path /path/to/alcp/lib -provider libopenssl-compat \
-evp aes-128-gcm

openssl speed -provider-path /path/to/alcp/lib -provider libopenssl-compat \
-evp aes-192-gcm
```

```
openssl speed -provider-path /path/to/alcp/lib -provider libopenssl-compat \
-evp aes-256-gcm
```

Part V

Appendix

Chapter 4

Compiling OpenSSL from the source.

```
git clone https://github.com/openssl/openssl.git -b openssl-3.0.5
cd openssl
./Configure --prefix=/usr/local
make -j $(nproc --all)
sudo make install
```