Part 4. 付録

付録１。プログラミング環境

4.1.1 ビルド方法

4.1.2 デバッグツール

1) デバッグメッセージ

wolfSSLは内部でのデバッグログを標準エラー出力に出力させるオプションを用意しています。この機能を有効にするためには、wolfSSLをDEBUG_WOLFSSLマクロまたは--enable-degubオプションを与えてビルドしておく必要があります。さらに、実行時にプログラム中から以下の関数を呼び出して出力を許可します。また、プログラム中でログ出力を停止したい場合はwolfSSL_Debugging_OFF()を呼び出してログ出力を停止することもできます。

```
wolfSSL_Debugging_ON();
```

デバッグメッセージはデフォルトでは標準エラー出力に出力されます。標準出力に出力したい場合はコンフィグレーションオプションにWOLFSSL_LOG_PRINTFを指定します。

また、組み込み環境などでターミナルなどにメッセージを出力できない場合に特別に確保したメモリーバッファーに出力したい場合や独自のヘッダーを付加したり独自のフォーマットで出力したい場合などはユーザ独自の出力関数を定義して使用することができます。その場合は、マクロ名のWOLFSSL_USER_LOGでユーザ独自関数の名前を定義します。この関数は出力すべきメッセージを文字列アーギュメントとして受け付けるようにします。

#define WOLFSSL_USER_LOG myPrint

int myPrint(char *msg);

2) TLSレコードの復号

TLSメッセージは、WireSharkなどのパケットキャプチャアプリを使うとネットワークパケットとして取得、解析できますがTLSパケットの内容は暗号化されています。OpenSSLやwolfSSLなどのTLSライブラリではパケットの復号に必要な情報をファイルとして出力するための手段を提供しています。パケットキャプチャアプリにこのファイルを設定することによりパケットを復号して表示させることができます。復号されるパケットはTLSのハンドシェークパケットだけでなく、ハンドシェーク後のアプリケーションデータも含まれます。

2-1) KeyLogファイルパスの定義
KeyLogファイルを使用する場合には次の2行をアプリケーション内の先頭付近に追加します。

```
#define SSLKEYLOGFILE  "./MyKeyLog.txt"
#include "example_common.h"
```

example_common.h（Examples/include/example_common.hを参照）にKeylogコールバック関数が既に実装されています。インクルード文の直前の上記#defineによってKeylogコールバック関数が有効になります。

2-2) KeyLogコールバック関数の登録

プログラムでは有効になったKeyLogコールバック関数を以下の様にして登録します。

```
SSL_CTX_set_keylog_callback(ctx, MyKeyLog_cb);
```

プログラムの実行に伴って、KeyLogファイルが作成され内容が追記されていきます。従って、そのままではファイルは大きくなり続けるので、適宜ファイルの内容を切り詰めるか、ファイルを削除します。

2-3) KeyLogファイルのWireSharkへの登録

WireSharkを起動したらメニューバーの"編集" > "設定" > Protocols" > "TLS"　と辿っていくと"Transport Layer Security" 設定画面になります。この画面の一番下に"(Pre)-Master-Secret log filename"の 設定欄があります。"Browse..."ボタンを押して、前述のKeyLogファイルを指定します。

2-4) データの復号と表示

ハンドシェークとアラートメッセージはパケットの概要ペインや個別パケット表示のペインにデフォルトで復号されます。一方、アプリケーションデータは最下ペインに表示される16進とASCII表示で内容を確認することになります。その際にそのペインのさらに下に"Frame"タブと"Decrypted TLS"タブが存在していること注意してください。デフォルトでは"Frame"タブが選択されているので暗号化されたデータが表示されています。復号されたアプリケーションデータを表示するには"Decrypted TLS"タブを選択します。

3) ヒープ使用状況

```
WOLFSSL_TRACK_MEMORYの使い方
```

4) テスト用証明書、鍵

certs下ファイルの使い方 test_certs.hの使い方