特設 HTTPS & TLSのしくみ

第1章 数百行のシンプルTLSクライアントで学ぶ

実装して理解するセキュリティ・プロトコルTLS

古城 隆

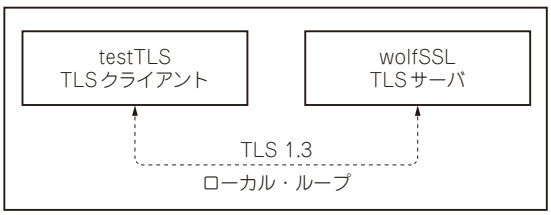

図1 最初に試すネットワーク環境

リスト1 Python環境のインストール

```
$ cd <YOUR_WORK_ROOT>
$ sudo apt update
$ sudo apt install python3-pip python3-dev
                                libffi-dev python3-venv
$ python3 -m venv tlsenv
$ source tlsenv/bin/activate
$ pip3 install --upgrade pip
$ pip3 install cffi tox pytest
$ sudo apt install git build-essential libssl-dev
                                automake libtool
```

TLS（Transport Layer Security）は，インターネットなどのネットワーク上で通信を行う際に，データの盗聴や改ざん，なりすましを防ぐための暗号化プロトコルです．インターネットを使うならTLSの世話にならない日はないでしょう．

本稿では簡単なTLSクライアントをPythonで作り，TLSプロトコルを動作させながらその仕組みを見ていきます．

セキュリティ・プロトコルの代表選手TLS 1.3

TLSは2018年に大幅に見直しが行われました．TLS 1.3として整理され，安全性はもちろんスループットも大きく改善されています．反面，通信の開始時に行われるハンドシェイクの大半が暗号化されて，勉強のために中身をのぞいてみたり，自分でプロトタイプを作ってみたりするというようなことは難しくなってしまいました．

とはいえ，プロトコルの構造はきちんと整理されているので，正しく理解すれば，数百行のPythonプログラムで基本的な機能を動作させて，本物のTLSサーバと通信させることが可能です．本稿ではそんなシンプルなTLSクライアントを実際に動かします．

テスト環境の構築

今回作るTLSクライアントは，インターネット（TCP/IP）に接続可能でありさえすればTLSサーバと通信できます．ここでは手軽に実験するため，**図1**のようにTLSクライアントとTLSサーバを同じPC上に構築し，互いに通信させます．クライアントは基本的にはPythonが動作する環境ならどこでも試すことができます．

プログラムは次のURLからダウンロードできます．

```
https://www.cqpub.co.jp/interface/download/
https://github.com/kojo1/InterfaceTLS/tree/master
```

● Python環境のセットアップ

Python 3と暗号ライブラリであるwolfcrypt-pyをインストールします．Python 3は既にインストールしてあれば不要です．Python環境がインストールされていない場合は**リスト1**のようにインストールします．

● wolfSSLとwolfcrypt-pyのセットアップ

wolfSSLは，主に組み込みシステムやIoTデバイス向けに開発された軽量で高速なSSL/TLS（Secure Sockets Layer/Transport Layer Security）ライブラリです．

wolfcrypt-pyはwolfSSLの上で動作するPython向け暗号ライブラリです．最新版のwolfcrypt-pyをローカル・ビルドして使います．通信テストのために建てるサーバも，wolfSSLに含まれるものを使います．

wolfSSL，wolfcrypt-pyはいわゆるデュアルライセンスのソフトウェアです．商用製品などに組み込んで