Secure Socket Layer (SSL) for Micro-Controller over Wireless LAN

This document illustrates how to secure network link by using SSL. The example setups a SSL connection with Apache Web server, and transmit/receive data securely via crypto processing.

Table of Contents

1 Introduction

This document illustrates how to secure network link by using SSL. PolarSSL 1.3.3 is used to support SSL connection. A SSL client sample codes is provided to setups a SSL connection with Apache Web server, and transmit/receive data securely via crypto processing. Following sections explain how to execute the SSL client sample codes and all the provided PolarSSL configuration files which can be used with the SSL client.

2 SSL Client Configuration

A SSL client sample is already implemented in ssl_client.c to demonstrate the SSL connection. The SSL client can be executed by interactive mode command. To support this SSL client command in interactive mode, the definitions of CONFIG_INTERACTIVE_MODE in main.c and CONFIG_SSL_CLIENT in wifi_interactive_mode.c must be modified as following.

```
/* main.c */
#define CONFIG_INTERACTIVE_MODE        1

/* wifi_interactive_mode.c */
#define CONFIG_SSL_CLIENT              1
```

By specifying an IP address in command, the SSL client will start to connect the SSL server related this address via HTTPS on port 443. The following is the ssl_client command listed in interactive mode.

```
# help
COMMAND LIST:
==============================
    wifi_connect
    wifi_disconnect
    wifi_info
    wifi_on
    wifi_off
    wifi_ap
    wifi_scan
    wifi_get_rssi
    iwpriv
    wifi_promisc
    wifi_simple_config
    wifi_wps
    ssl_client
    ttcp
    ping
    exit
    help

# ssl_client
Usage: ssl client SSL SERVER HOST
```

3 PolarSSL Configurations

Some configuration files are provided to configure PolarSSL library to support different cipher suites. The following sub-sections explain the PolarSSL configurations for all cipher suites, RSA-AES-SHA cipher suites, and SRP-AES-SHA cipher suites.

3.1.1 Configuration for All Cipher Suites

PolarSSL library provides several cipher suites for SSL connection. The provided configuration file of config_all.h enables all the supported cipher suites of PolarSSL 1.3.3. Notify that SRP key exchange is not implemented by PolarSSL 1.3.3. Therefore, cipher suites using SRP key exchange will not be proposed in SSL handshake if the config_all.h configuration is adopted. To enable config_all.h configuration for PolarSSL library, the definitions in polarssl/config.h should be modified as following. However, this configuration may not be suitable to execute SSL client due to platform memory limitation when WLAN driver is also used

```
/* polarssl/config.h */
#define CONFIG_SSL_RSA      0
#define CONFIG_SSL_SRP      0
```

The following is the executing of SSL client when configuring PolarSSL to use config_all.h. In this example, SSL server determines to use DHE-RSA-AES-256-GCM-SHA384 from all the proposed cipher suites during SSL handshake. SSL client requests to get homepage of web server via SSL connection successfully.

```
# ssl_client 192.168.13.15

#
  . Connecting to tcp/192.168.13.15/443... ok

  . Setting up the SSL/TLS structure... ok

  . Performing the SSL/TLS handshake... ok

  . Use ciphersuite TLS-ECDHE-RSA-WITH-AES-256-GCM-SHA384

  > Write to server: 18 bytes written

GET / HTTP/1.0



  < Read from server: 269 bytes read

HTTP/1.1 200 OK
```

3.1.2 Configuration for RSA-AES-SHA Cipher Suite

To reduce the memory requirement when using PolarSSL library, a configuration file of config_rsa.h is provided to only enable all the cipher suites of RSA-AES-SHA. Therefore, only cipher suites using RSA-AES-SHA will be proposed in SSL handshake if the config_rsa.h configuration is adopted. To enable config_rsa.h configuration for PolarSSL library, the definitions in polarssl/config.h should be modified as following.

```
/* polarssl/config.h */
#define CONFIG_SSL_RSA      1
#define CONFIG_SSL_SRP      0
```

The following is the executing of SSL client when configuring PolarSSL to use config_rsa.h. In this example, SSL server in 192.168.13.15 is connected by ssl_client command. SSL server determines to use RSA-AES-256-CBC-SHA256 from all the proposed RSA-AES-SHA cipher suites during SSL handshake. SSL client requests to get homepage of web server via SSL connection successfully.

```
# ssl_client 192.168.13.15

#
  . Connecting to tcp/192.168.13.15/443... ok

  . Setting up the SSL/TLS structure... ok

  . Performing the SSL/TLS handshake... ok

  . Use ciphersuite TLS-RSA-WITH-AES-256-CBC-SHA256

  > Write to server: 18 bytes written

GET / HTTP/1.0



  < Read from server: 269 bytes read

HTTP/1.1 200 OK
```

3.1.3 Configuration for SRP-AES-SHA Cipher Suite

To support SSL connection without server certificate, SRP (Secure Remote Password) [RFC5054] key exchange is implemented to add to the original PolarSSL 1.3.3 library. The configuration file of config_srp.h is provided to only enable the cipher suite of SRP-AES-256-CBC-SHA (Kx=SRP, Au=None, Enc=AES(256), Mac=SHA1). Therefore, only this cipher suite will be proposed in SSL handshake if the config_srp.h configuration is adopted. To enable config_srp.h configuration for PolarSSL library, the definitions in polarssl/config.h should be modified as following. SSL_USE_SRP in ssl_client.c should also be enabled to be able to input SRP username and password in ssl_client command.

```
/* polarssl/config.h */
#define CONFIG_SSL_RSA      0
#define CONFIG_SSL_SRP      1

/* ssl_client.c */
#define SSL_USE_SRP         1
```

Although server certificate is not used by SRP key exchange, SRP username and password should be setup for SSL handshake by SSL client to match the setting in SSL server.

The following is the executing of SSL client when configuring PolarSSL to use config_srp.h. In this example, username is set to “test”, and password is set to “12345678”. SSL server determines to use the only SRP-AES-256-CBC-SHA cipher suite during SSL handshake. SSL client

requests to get homepage of web server via SSL connection successfully. Notify that SRP is available in Apache 2.4.4 and later.

```
# ssl_client 192.168.13.15 test 12345678

#
  . Connecting to tcp/192.168.13.15/443... ok

  . Setting up the SSL/TLS structure... ok

  . Performing the SSL/TLS handshake... ok

  . Use ciphersuite TLS-SRP-WITH-AES-256-CBC-SHA

  > Write to server: 18 bytes written

GET / HTTP/1.0



  < Read from server: 269 bytes read

HTTP/1.1 200 OK
```

4 Memory Configuration

The config_all.h using default PolarSSL library configuration will require large heap memory for SSL input/output buffer (16384 bytes) and big number (512 bytes). However, this configuration will not be enough to execute SSL client on the platform with 128k bytes memory when WLAN driver is also used. To reduce the memory usage, the definition of POLARSSL_CONFIG_OPTIONS in config_rsa.h and config_srp.h is enabled to use customized configuration. The definitions of SSL_MAX_CONTENT_LEN for SSL input/output buffer and POLARSSL_MPI_MAX_SIZE for big number size are modified according to the requirement of RSA and SRP cipher suites as following. Notify that the setting of config_srp.h is only for the SSL server using the 1024-bit SRP group parameter. It should be increased based on server requirement.

```
/* polarssl/config_rsa.h */
#define POLARSSL_CONFIG_OPTIONS
#define POLARSSL_MPI_MAX_SIZE        256
#define SSL_MAX_CONTENT_LEN          768

/* polarssl/config_srp.h */
#define POLARSSL_CONFIG_OPTIONS
#define POLARSSL_MPI_MAX_SIZE        128
#define SSL_MAX_CONTENT_LEN          512
```

Beside of heap usage, task stack size should also be considered when using PolarSSL library. For example, the definition of POLARSSL_DEBUG_C enabled in config_all.h, config_rsa.h and config_srp.h will enable debug functions of PolarSSL library, but it will also require more task stack size. It will increase about 1k bytes of stack size for config_rsa.h configuration compared with that when disabling POLARSSL_DEBUG_C. Therefore, STACKSIZE of SSL client task could be modified based on the PolarSSL configuration.

```
/* ssl_client.c */
#define STACKSIZE     1150

/* config_all.h, config_rsa.h, config_srp.h */
#define POLARSSL_DEBUG_C
```