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

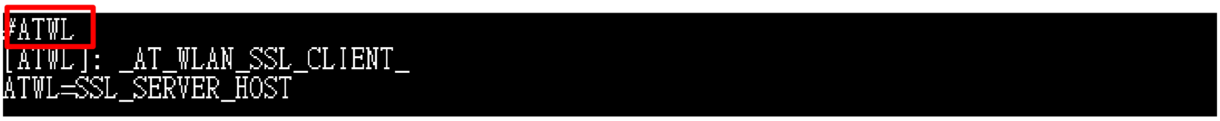
This document illustrates how to secure network link by using SSL. PolarSSL 1.3.8 is used to support SSL connection. A SSL client sample codes is provided to setups a SSL connection with Apache Web server, and transmit/receive data securely via crypto processing. Following sections explain how to execute 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 AT command, the definitions of CONFIG_SSL_CLIENT in platform_opts.h must be modified as following.

```
/* platform_opts.h */
#define CONFIG_SSL_CLIENT                    1
```

By specifying an IP address in ATWL command, the SSL client will start to connect the SSL server related this address via HTTPS on port 443. The following is the information of ATWL command.



```
#ATWL
[ATWL]: _AT_WLAN_SSL_CLIENT_
ATWL=SSL_SERVER_HOST
```

3 PolarSSL Configurations

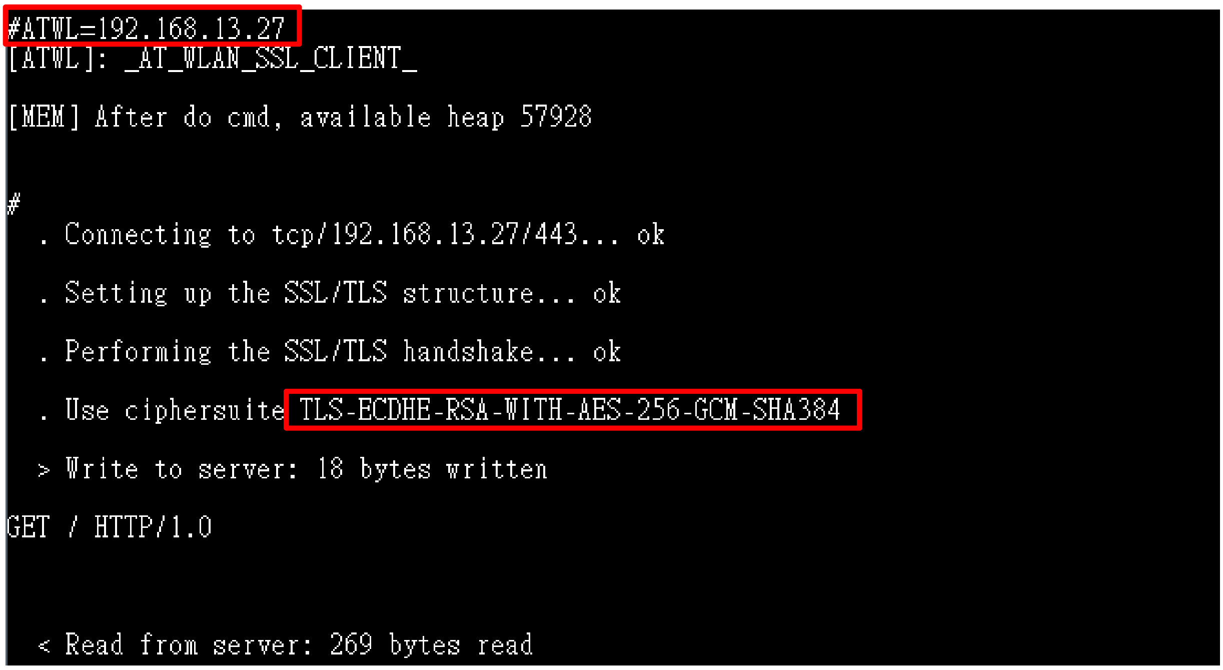
Some configuration files are provided to configure PolarSSL library to support different cipher suites. The following sub-sections explain the PolarSSL configurations for all cipher suites and RSA-AES-SHA cipher suites.

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 1.3.8. To enable config_all.h configuration for PolarSSL library, the definitions in polarssl/config.h should be modified as following.

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

The following is the executing of SSL client when configuring PolarSSL to use config_all.h. In this example, SSL server in 192.168.13.27 is connected by ATWL command. SSL server determines to use TLS-ECDHE-RSA-WITH-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.



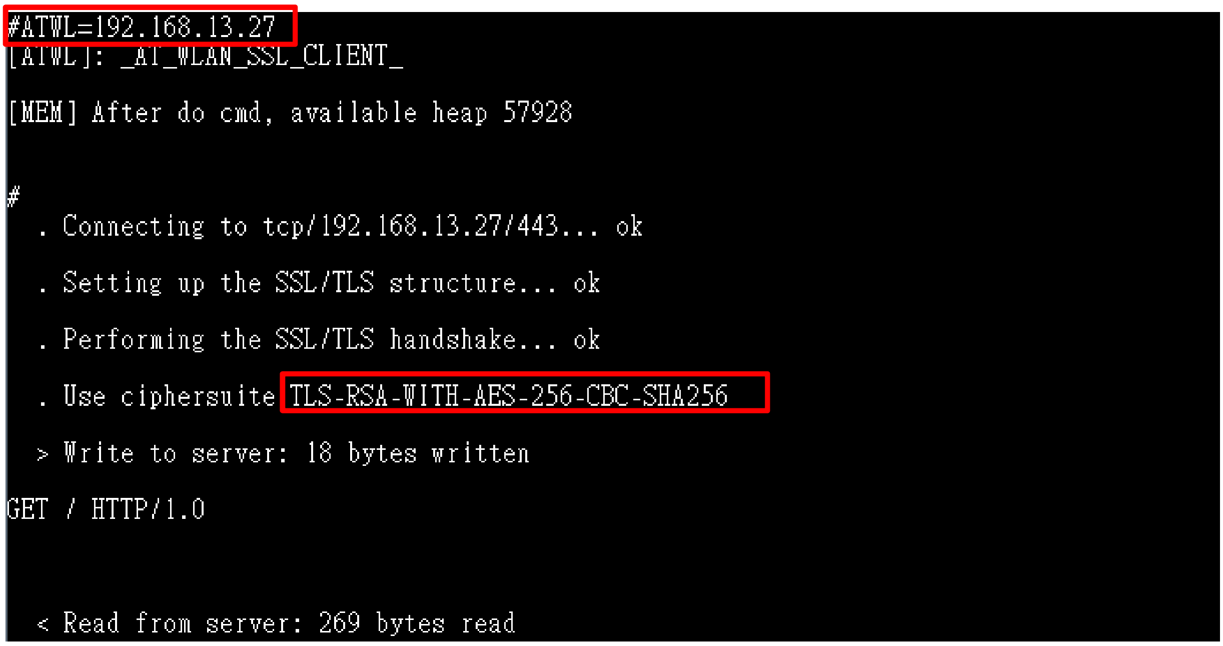
```
#ATWL=192.168.13.27
[ATWL]: _AT_WLAN_SSL_CLIENT_

[MEM] After do cmd, available heap 57928


#
  . Connecting to tcp/192.168.13.27/443... 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
```

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
```

The following is the executing of SSL client when configuring PolarSSL to use config_rsa.h. In this example, SSL server in 192.168.13.27 is connected by ATWL command. SSL server determines to use TLS-RSA-WITH-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.



```
#ATWL=192.168.13.27
[ATWL]: _AT_WLAN_SSL_CLIENT_

[MEM] After do cmd, available heap 57928


#
  . Connecting to tcp/192.168.13.27/443... 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
```

4 Memory Configuration

The config_all.h using default PolarSSL library configuration will require large heap memory for SSL input/output buffer (16384 bytes). To reduce the memory usage, customized configuration is enabled in config_rsa.h. The definitions of SSL_MAX_CONTENT_LEN for SSL input/output buffer is modified according to the requirement of RSA cipher suites as following. This value may need to be increased based on the cipher suite determined by server or the size of data transferred from server.

```
/* polarssl/config_rsa.h */
#define SSL_MAX_CONTENT_LEN            4096
```

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 and config_rsa.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 */
#define POLARSSL_DEBUG_C
```