GENERAL NOTES

SIMCOM OFFERS THIS INFORMATION AS A SERVICE TO ITS CUSTOMERS, TO SUPPORT APPLICATION AND ENGINEERING EFFORTS THAT USE THE PRODUCTS DESIGNED BY SIMCOM. THE INFORMATION PROVIDED IS BASED UPON REQUIREMENTS SPECIFICALLY PROVIDED TO SIMCOM BY THE CUSTOMERS. SIMCOM HAS NOT UNDERTAKEN ANY INDEPENDENT SEARCH FOR ADDITIONAL RELEVANT INFORMATION, INCLUDING ANY INFORMATION THAT MAY BE IN THE CUSTOMER’S POSSESSION. FURTHERMORE, SYSTEM VALIDATION OF THIS PRODUCT DESIGNED BY SIMCOM WITHIN A LARGER ELECTRONIC SYSTEM REMAINS THE RESPONSIBILITY OF THE CUSTOMER OR THE CUSTOMER’S SYSTEM INTEGRATOR. ALL SPECIFICATIONS SUPPLIED HEREIN ARE SUBJECT TO CHANGE.

COPYRIGHT

THIS DOCUMENT CONTAINS PROPRIETARY TECHNICAL INFORMATION WHICH IS THE PROPERTY OF SIMCOM WIRELESS SOLUTIONS LIMITED. COPYING, TO OTHERS AND USING THIS DOCUMENT, ARE FORBIDDEN WITHOUT EXPRESS AUTHORITY BY SIMCOM. OFFENDERS ARE LIABLE TO THE PAYMENT OF INDEMNIFICATIONS. ALL RIGHTS RESERVED BY SIMCOM IN THE PROPRIETARY TECHNICAL INFORMATION, INCLUDING BUT NOT LIMITED TO REGISTRATION GRANTING OF A PATENT, A UTILITY MODEL OR DESIGN. ALL SPECIFICATION SUPPLIED HEREIN ARE SUBJECT TO CHANGE WITHOUT NOTICE AT ANY TIME.

SIMCom Wireless Solutions Limited
Building B, SIM Technology Building, No.633 Jinzhong Road, Changning District, Shanghai P.R. China
Tel: +86 21 31575100
Email: simcom@simcom.com

For more information, please visit:
https://www.simcom.com/download/list-863-en.html

For technical support, or to report documentation errors, please visit:
https://www.simcom.com/ask/ or email to: support@simcom.com

Copyright © 2020 SIMCom Wireless Solutions Limited All Rights Reserved.
About Document

Version History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Owner</th>
<th>What is new</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1.00</td>
<td>2020.08.11</td>
<td>Wei.Zhang</td>
<td>All</td>
</tr>
</tbody>
</table>

Scope

This document applies to the following products

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size(mm)</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIM7080G</td>
<td>CAT-M/NB</td>
<td>17.6*15.7 *2.3</td>
<td>N/A</td>
</tr>
<tr>
<td>SIM7070G/SIM7070E</td>
<td>CAT-M/NB/GPRS</td>
<td>24<em>24</em>2.4</td>
<td>N/A</td>
</tr>
<tr>
<td>SIM7070G-NG</td>
<td>NB/GPRS</td>
<td>24<em>24</em>2.4</td>
<td>N/A</td>
</tr>
<tr>
<td>SIM7090G</td>
<td>CAT-M/NB</td>
<td>14.8<em>12.8</em>2.0</td>
<td>N/A</td>
</tr>
</tbody>
</table>
## Contents

About Document .................................................................................................................................. 3  
Version History ................................................................................................................................... 3  
Scope .................................................................................................................................................. 3  

Contents ............................................................................................................................................... 4  

1 Introduction ...................................................................................................................................... 5  
   1.1 Purpose of the document ............................................................................................................... 5  
   1.2 Related documents ....................................................................................................................... 5  
   1.3 Conventions and abbreviations .................................................................................................... 5  

2 SSL Introduction .............................................................................................................................. 7  
   2.1 SSL Versions and Cipher Suites .................................................................................................. 7  
   2.2 Supported Certificate format ........................................................................................................ 9  

3 AT Commands that support SSL ..................................................................................................... 10  

4 Certificate Management .................................................................................................................. 11  
   4.1 Configure SSL parameters ........................................................................................................... 11  
   4.2 Import and Convert Root CA ....................................................................................................... 12  
   4.3 Import and Convert Certificate .................................................................................................... 13  
   4.4 Import and Convert PSK .............................................................................................................. 14
1 Introduction

1.1 Purpose of the document

Based on module AT command manual, this document will introduce SSL application process.

The following applications of SIM7070_SIM7080_SIM7090 series module support SSL: HTTP, FTP, TCPUDP, MQTT and EMAIL.

Developers could understand and develop application quickly and efficiently based on this document.

1.2 Related documents

[2] SIM7070_SIM7080_SIM7090 Series_FS_Application Note
[3] SIM7070_SIM7080_SIM7090 Series_HTTP(S)_Application Note
[4] SIM7070_SIM7080_SIM7090 Series_FTP(S)_Application Note
[5] SIM7070_SIM7080_SIM7090 Series_TCPUDP(S)_Application Note
[6] SIM7070_SIM7080_SIM7090 Series_MQTT(S)_Application Note
[7] SIM7070_SIM7080_SIM7090 Series_Email_Application Note
[8] RFC7925
[9] RFC2246

1.3 Conventions and abbreviations

In this document, the GSM engines are referred to as following term:

- ME (Mobile Equipment);
- MS (Mobile Station);
- TA (Terminal Adapter);
- DCE (Data Communication Equipment) or facsimile DCE (FAX modem, FAX board);

In application, controlling device controls the GSM engine by sending AT Command via its serial interface. The controlling device at the other end of the serial line is referred to as following term:
- TE (Terminal Equipment);
- DTE (Data Terminal Equipment) or plainly "the application" which is running on an embedded system;
2 SSL Introduction

- SSL (Secure Sockets Layer), a security protocol. It was put forward by Netscape in the first version of Web browser. The aim is to provide security and data integrity for network communications. SSL encrypts the network connections at the transport layer.

- SSL uses public key technology to ensure the confidentiality and reliability of communication between two applications and to ensure that communication between client and server applications is not eavesdropped by attackers. It can be supported at both ends of the server and client, and has become an industrial standard for secure communication over the Internet. Current Web browsers generally combine HTTP and SSL to achieve secure communication. This Agreement and its successor are TLS (Transport Layer Security, TLS).

- TLS uses key algorithm to provide endpoint authentication and communication security on the Internet. It is based on the public key infrastructure. In typical implementations, however, only the network server is authenticated reliably, while the client is not necessarily. This is because the public key infrastructure is generally commercial, and electronic signature certificates usually need to be paid for. The protocol is designed to enable master-slave architecture application communication itself to prevent tapping, tampering, and message forgery

- DTLS (Datagram Transport Layer Security) is the data transmission layer security protocol. TLS cannot be used to ensure the security of the data transmitted on UDP, so Datagram TLS extends the existing TLS protocol architecture to support UDP, that is datagram transmission. DTLS 1.0 is based on TLS 1.1, and DTLS 1.2 is based on TLS 1.2.

- DTLS, TLS and SSL encrypt network connections at the transport layer, DTLS is above the UDP transport protocol, and TLS is above the TCP transport protocol.

2.1 SSL Versions and Cipher Suites

The following SSL versions are supported.

<table>
<thead>
<tr>
<th>Version</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>TLS1.0</td>
<td></td>
</tr>
<tr>
<td>TLS1.1</td>
<td></td>
</tr>
<tr>
<td>TLS1.2</td>
<td></td>
</tr>
<tr>
<td>DTLS1.0</td>
<td></td>
</tr>
</tbody>
</table>
DTLS1.2

The following table shows SSL cipher suites supported by SIM7070_SIM7080_SIM7090 series module. For detailed description of cipher suites, please refer to RFC 2246-The TLS Protocol Version 1.0.

<table>
<thead>
<tr>
<th>Code of Cipher Suites</th>
<th>Name of Cipher Suites</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x008A</td>
<td>PSK_WITH_RC4_128_SHA</td>
</tr>
<tr>
<td>0x008B</td>
<td>PSK_WITH_3DES_EDE_CBC_SHA</td>
</tr>
<tr>
<td>0x008C</td>
<td>PSK_WITH_AES_128_CBC_SHA</td>
</tr>
<tr>
<td>0x008D</td>
<td>PSK_WITH_AES_256_CBC_SHA</td>
</tr>
<tr>
<td>0x00A8</td>
<td>PSK_WITH_AES_128_GCM_SHA256</td>
</tr>
<tr>
<td>0x00A9</td>
<td>PSK_WITH_AES_256_GCM_SHA384</td>
</tr>
<tr>
<td>0x00AE</td>
<td>PSK_WITH_AES_128_CBC_SHA256</td>
</tr>
<tr>
<td>0x00AF</td>
<td>PSK_WITH_AES_256_CBC_SHA384</td>
</tr>
<tr>
<td>0xC0A8</td>
<td>PSK_WITH_AES_128_CCM_8</td>
</tr>
<tr>
<td>0x002F</td>
<td>RSA_WITH_AES_128_CBC_SHA</td>
</tr>
<tr>
<td>0x0033</td>
<td>DHE_RSA_WITH_AES_128_CBC_SHA</td>
</tr>
<tr>
<td>0x0035</td>
<td>RSA_WITH_AES_256_CBC_SHA</td>
</tr>
<tr>
<td>0x0039</td>
<td>DHE_RSA_WITH_AES_256_CBC_SHA</td>
</tr>
<tr>
<td>0x003C</td>
<td>RSA_WITH_AES_128_CBC_SHA256</td>
</tr>
<tr>
<td>0x003D</td>
<td>RSA_WITH_AES_256_CBC_SHA256</td>
</tr>
<tr>
<td>0x0067</td>
<td>DHE_RSA_WITH_AES_128_CBC_SHA256</td>
</tr>
<tr>
<td>0x006B</td>
<td>DHE_RSA_WITH_AES_256_CBC_SHA256</td>
</tr>
<tr>
<td>0x009C</td>
<td>RSA_WITH_AES_128_GCM_SHA256</td>
</tr>
<tr>
<td>0x009D</td>
<td>RSA_WITH_AES_256_GCM_SHA384</td>
</tr>
<tr>
<td>0x009E</td>
<td>DHE_RSA_WITH_AES_128_GCM_SHA256</td>
</tr>
<tr>
<td>0x009F</td>
<td>DHE_RSA_WITH_AES_256_GCM_SHA384</td>
</tr>
<tr>
<td>0xC004</td>
<td>ECDH_ECDSA_WITH_AES_128_CBC_SHA</td>
</tr>
<tr>
<td>0xC005</td>
<td>ECDH_ECDSA_WITH_AES_128_CBC_SHA</td>
</tr>
<tr>
<td>0xC009</td>
<td>ECDHE_ECDSA_WITH_AES_128_CBC_SHA</td>
</tr>
<tr>
<td>0xC00A</td>
<td>ECDHE_ECDSA_WITH_AES_256_CBC_SHA</td>
</tr>
<tr>
<td>0xC00E</td>
<td>ECDH_RSA_WITH_AES_128_CBC_SHA</td>
</tr>
<tr>
<td>0xC00F</td>
<td>ECDH_RSA_WITH_AES_256_CBC_SHA</td>
</tr>
<tr>
<td>0xC013</td>
<td>ECDH_RSA_WITH_AES_128_CBC_SHA</td>
</tr>
<tr>
<td>0xC014</td>
<td>ECDH_RSA_WITH_AES_256_CBC_SHA</td>
</tr>
<tr>
<td>0xC023</td>
<td>ECDHE_ECDSA_WITH_AES_128_CBC_SHA256</td>
</tr>
<tr>
<td>0xC024</td>
<td>ECDHE_ECDSA_WITH_AES_256_CBC_SHA256</td>
</tr>
<tr>
<td>0xC025</td>
<td>ECDH_ECDSA_WITH_AES_128_CBC_SHA256</td>
</tr>
<tr>
<td>0xC026</td>
<td>ECDH_ECDSA_WITH_AES_256_CBC_SHA256</td>
</tr>
<tr>
<td>0xC027</td>
<td>ECDH_RSA_WITH_AES_128_CBC_SHA256</td>
</tr>
<tr>
<td>0xC028</td>
<td>ECDH_RSA_WITH_AES_256_CBC_SHA256</td>
</tr>
</tbody>
</table>
2.2 Supported Certificate format

- SSL Certificates is a file that uses digital encryption technology to encrypt the details of the publisher's information.

- SSL certificates are encoded in the binary format specified in the X.509 ITU-T standard, which is a well-established telecom standard for storing structured binary data. The binary X.509 data is sometimes stored as raw binary data and sometimes encoded using Base64 encoding. Several X.509 encoding formats exists. We can handle the following formats: .PEM, .DER, and .P7B.

- Typically, a certificate's private key and public certificate are stored as two .PEM encoded files. Root certificates (CAs) are typically stored in the container format .P7B, which can store multiple CAs. The root certificates are used by SSL function when certifying ( authenticating) the peer side.
# 3 AT Commands that support SSL

The module provides AT commands that can be used by device terminals as follows:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+CSSLCFG</td>
<td>Configure SSL parameters of a context identifier</td>
</tr>
<tr>
<td>AT+CASSLCFG</td>
<td>Set SSL Certificate and Timeout Parameters(TCP/UDP)</td>
</tr>
<tr>
<td>AT+SHSSL</td>
<td>Select SSL Configure(HTTP)</td>
</tr>
<tr>
<td>AT+FTPSSL</td>
<td>Select FTP SSL Configure</td>
</tr>
<tr>
<td>AT+SMSSL</td>
<td>Select SSL Configure</td>
</tr>
<tr>
<td>AT+EMAILSSL</td>
<td>Set Email SSL function</td>
</tr>
</tbody>
</table>

For detail information, please refer to "SIM7070_SIM7080_SIM7090 Series_AT Command Manual".
# 4 Certificate Management

When SSL establishes communication, it is necessary to verify the identity of both sides of the communication, which is divided into one-way authentication and two-way authentication.

One way authentication is the client to verify the certificate of the server. The server sends the server certificate to the client. The client verifies that the root certificate that issued the server certificate is trustworthy, and if so continues the communication process.

After the two-way authentication client verifies the server certificate, the client needs to send its own certificate to the server and let the server verify its client certificate. The validation process is the same, all need to confirm whether the root certificate of the certificate can be trusted.

Module uses its own binary format for storing certificates, a format optimized for speed and size. We provide AT command that can convert standard .PEM, .DER, and .P7B formats into its internal format.

By default, SSL only performs server-side authentication, client-side authentication is optional.

The following example is to visit Baidu web as an example.

## 4.1 Configure SSL parameters

Module can support the existence of 6 configuration files with sequence numbers from 0-5.

The following example SSL configuration will take the second configuration file as an example.

```
//Example of configure SSL parameters.
AT+CSSLCFG="SSLVERSION",1,3 //Set the protocol type of SSL version
1 means the second configuration file.
3 means TLS1.2

OK
AT+CSSLCFG="CIPHERSUITE",1,0,0x009c //Configure the ciphersuite.
1 means the second configuration file.
0 means cipher_index.
0x009c means
TLS_RSA_WITH_AES_128_GCM_SHA256.
You can check it from the Supported SSL Cipher Suites table in SSL Versions and Cipher Suites
```
You can choose not to fill, and it will automatically default. Otherwise, select at least 1 cipher suite or at most 8 cipher suites. Range:0-7 This is an option.

**OK**

**AT+CSSLCFG=“CIPHERSUITE”,1,7,0x002f**

//Configure SNI
1 means the second configuration file. www.baidu.com is severname.
SNI (Server Name Indication) is an extension of TLS and is used to solve the situation where a server has multiple domain names. This is an option.

**OK**

**AT+CSSLCFG=“SNI”,1,www.baidu.com**

**OK**

**AT+CSSLCFG=“IGNORERTCTIME”,1,1**

//Configure whether to ignore time.
The first 1 means the second configuration file.
The second 1 means ignore the RTC time.
If it's 0 that means do not ignore the RTC time. This is an option.

**OK**

**AT+CSSLCFG=“CTXINDEX”,1**

//Query all the parameters that have been set. This is an option.

```
+CSSLCFG:
3,0x009c,0x0035,0xc0a8,0xc030,0xc0a8,0xc013,0
x009d,0x002f,1,""
```

**OK**

## 4.2 Import and Convert Root CA

A root CA certificate can be used by module during the initial SSL handshake. The root CA is then used as a trusted third party and module uses the root CA for certifying the peer's certificate. Certifying the peer side is optional in Module, however, if used, the peer side must provide a certificate signed by a trusted root CA as part of the SSL handshake.

The root CA certificate(s) must be stored on a file system. In many cases, only one root CA certificate is required in the device. The same root CA can be used to sign all certificates the device must verify. The below example programs show how to store and use one root CA.
4.3 Import and Convert Certificate

//Example of import and convert certificate.
AT+CFSINIT
OK
AT+CFSWFILE=3,Client.cer,0,1024,10000
OK
AT+CFSTERM
OK

NOTE

Perform the above steps at least once. As long as the certificate is not updated, there is no need to repeat the operation.
OK
AT+CFSWFILE=3,Client_key.pem,0,512,10000

//Import Client_key.pem
3 means this file will put in the “customer” directory.
Client_key.pem is private key that will be written to “customer” directory.
0 is starting writing point,512 is the size of private key,10000 means that you must complete it in 10000 ms.

OK
AT+CSSLCFG="CONVERT",1,Client.cer,Client_key.pem,"simcom"

//Convert Client.cer and Client_key.pem , and save it to file system.
1 means convert certificate. Client.cer is certificate. Client_key.pem is private key of Client.cer.
If Client_key.pem is encrypted,”simcom” is the corresponding password.

OK
AT+CFSDFILE=3,Client.cer

//After convert, delete temp Client.cer which is in “customer” directory because of security considerations.
3 means it belongs to the “customer” directory.
Client.cer is certificate name.

OK
AT+CFSDFILE=3,Client_key.pem

// After convert, delete temp Client_key.pem which is in “customer” directory because of security considerations.
3 means it belongs to the “customer” directory.
Client_key.pem is private key.

AT+CFSTEM
OK

NOTE
Perform the above steps at least once. As long as the certificate is not updated, there is no need to repeat the operation.

4.4 Import and Convert PSK

PSK(pre-shared key) is a series of keys that have been determined and known by both parties before
communication, and this series of keys relies on PSK Identify (referred to as PSK ID) for indexing.

After the client sends the Client Key Exchange with the PSK Client Params, the server will index the PSK ID to find the preset key (this key is also stored locally on the client), and then use an algorithm to combine this key and Parameters such as the random numbers of both parties generate the final symmetric key.

//Example of import and convert PSK.

AT+CFSINIT
OK
AT+CFSWFILE=3,psktable.secret,0,512,10000
OK
AT+CSSLCFG="CONVERT",3,psktable.secret
OK
AT+CFSDFILE=3,psktable.secret
OK
AT+CFSTERM
OK

//Initialize AT relate to file system functions

//Import psktable.secret.
3 means this file will put in the “customer” directory.
psktable.secret will be written to “customer” directory.
0 is starting writing point, 512 is the size of psktable.secret.
10000 means that you must complete it in 10 second.
The psktable (psktable.secret) format as follows:
<Identity>:<psk_key>
Identity and psk_key are corresponding.
Each Identity correspond to a psk_key.
For example:
user_zhao:313233
It should be noted that the Identity is string type and psk_key is hexadecimal string(e.g: If the psk_key is string “123”, you must write that “313233”)

//Configure the type of psktable to be converted, and 3 is psktable.

//After convert, delete temp psktable.secret which is in “customer” directory because of security considerations.
3 means it belongs to the “customer” directory.
psktable.secret is PSK file name.

//Release AT relates to file system functions.

NOTE

Perform the above steps at least once. As long as the certificate is not updated, there is no need to repeat the operation.