



# **SIM7500\_SIM7600\_SIM7800 Series\_SSL\_AT Command Manual\_V1.00**

**LTE Module**

**Shanghai SIMCom Wireless Solutions Ltd.**  
Building A, SIM Technology Building, No.633, Jinzhong Road  
Changning District 200335  
Tel:86-21-31575100/31575200  
support@simcom.com  
www.simcom.com

<b>Document Title:</b>	SIM7500_SIM7600_SIM7800	Series_SSL_AT	Command Manual
<b>Version:</b>	1.00		
<b>Date:</b>	2018-10-12		
<b>Status:</b>	Release		
<b>Document ID:</b>	SIM7500_SIM7600_SIM7800	Series_SSL_AT	Command Manual_V1.00

### 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 Limited., copying of this document and giving it to others and the using or communication of the contents thereof, are forbidden without express authority. Offenders are liable to the payment of damages. All rights reserved in the event of grant of a patent or the registration of a utility model or design. All specification supplied herein are subject to change without notice at any time.

Copyright © Shanghai SIMCom Wireless Solutions Ltd. 2018

## Version History

Version	Date	Chapter	What is new
V1.00	2018-09-28		New version

SIMCom Confidential

# Contents

<b>Version History .....</b>	<b>2</b>
<b>Contents.....</b>	<b>3</b>
<b>1 Introduction.....</b>	<b>4</b>
1.1 The SSL Context Management AT Commands .....	4
1.2 The process of Using SSL AT Commands .....	4
<b>2 Description of AT Command .....</b>	<b>5</b>
2.1 SSL Context Management AT .....	5
2.1.1 AT+CSSLCFG Configure the SSL Context.....	5
2.1.2 AT+CCERTDOWN Download certificate into the module.....	9
2.1.3 AT+CCERTLIST List certificates.....	9
2.1.4 AT+CCERTDELETE Delete certificates.....	10
2.2 SSL Services AT .....	10
2.2.1 AT+CCHSET Configure the report mode of sending and receiving data.....	10
2.2.2 AT+CCHMODE Configure the mode of sending and receiving data.....	11
2.2.3 AT+CCHSTART Start SSL service.....	12
2.2.4 AT+CCHSTOP Stop SSL service .....	13
2.2.5 AT+CCHADDR Get the IPv4 address.....	13
2.2.6 AT+CCHSSLCFG Set the SSL context .....	13
2.2.7 AT+CCHOPEN Connect to server.....	14
2.2.8 AT+CCHCLOSE Disconnect from server .....	15
2.2.9 AT+CCHSEND Send data to server .....	16
2.2.10 AT+CCHRECV Read the cached data that received from the server .....	17
2.3 Command result codes and unsolicited codes .....	18
2.3.1 Command result <err> codes.....	18
2.3.2 Unsolicited result codes.....	19
<b>3 Example.....</b>	<b>19</b>
3.1 Access to TCP server .....	20
3.2 Access to SSL/TLS server (not verify server and client).....	22
3.3 Access to SSL/TLS server (only verify the server) .....	24
3.4 Access to SSL/TLS server (verify server and client).....	26
3.5 Access to SSL/TLS server (only verify the client) .....	28
3.6 Access to SSL/TLS server in transparent mode.....	31
3.7 Download certificate into module.....	32

This document is a reference guide to all the AT commands defined for SSL. Through these SSL AT commands, you can communicate with a TCP or SSL server.

# 1 Introduction

## 1.1 The SSL Context Management AT Commands

**Step 1:** Configure SSL version by AT+CSSLCFG="sslversion",<ssl\_ctx\_index>,<sslversion>.

**Step 2:** Configure SSL authentication mode by AT+CSSLCFG="authmode",<ssl\_ctx\_index>,<authmode>.

**Step 3:** Configure the flag of ignore local time by

AT+CSSLCFG="ignorlocaltime",<ssl\_ctx\_index>,<ignoreltime>.

**Step 4:** Configure the max time in SSL negotiation stage by

AT+CSSLCFG="negotiatetime",<ssl\_ctx\_index>,<negotiatetime>.

**Step 5:** Configure the server root CA by AT+CSSLCFG="cacert",<ssl\_ctx\_index>,<ca\_file>.

**Step 6:** Configure the client certificate by AT+CSSLCFG="clientcert",<ssl\_ctx\_index>,<clientcert\_file>.

**Step 7:** Configure the client key by AT+CSSLCFG="clientkey",<ssl\_ctx\_index>,<clientkey\_file>.

**Step 8:** Download the certificate into the module by AT+CCERTDOWN.

**Step 9:** Delete the certificate from the module by AT+CCERTDELE.

**Step 10:** List the certificates by AT+CCERTLIST.

## 1.2 The process of Using SSL AT Commands

**Step 1:** Ensure GPRS network is available before performing SSL related operations.

**Step 2:** Configure the parameter of PDP context by AT+CGDCONT.

**Step 3:** Activate the PDP context to start SSL service by AT+CCHSTART.

**Step 4:** Configure SSL context by AT+CSSLCFG (if connect to SSL/TLS server).

**Step 5:** Set the SSL context used in SSL connection by AT+CCHSSLCFG (if connect to SSL/TLS server).

**Step 6:** Connect to the server by AT+CCHOPEN.

**Step 7:** Send data to the server by AT+CCHSEND.

**Step 8:** Receive data from server by AT+CCHRECV in manual receive mode.

**Step 9:** Disconnect from the server by AT+CCHCLOSE.

**Step 10:** Deactivate the PDP context to stop SSL service by AT+CCHSTOP.

**Note:**

## 2 Description of AT Command

### 2.1 SSL Context Management AT

#### 2.1.1 AT+CSSLCFG Configure the SSL Context

AT+CSSLCFG Configure the SSL Context	
<p>Test Command AT+CSSLCFG=?</p>	<p>Response</p> <pre>+CSSLCFG: "sslversion",(0-9),(0-4) +CSSLCFG: "authmode",(0-9),(0-3) +CSSLCFG: "ignorelocaltime",(0-9),(0,1) +CSSLCFG: "negotiatetime",(0-9),(10-300) +CSSLCFG: "cacert",(0-9),(5-128) +CSSLCFG: "clientcert",(0-9),(5-128) +CSSLCFG: "clientkey",(0-9),(5-128)</pre> <p>OK</p>
<p>Read Command AT+CSSLCFG?</p>	<p>Response</p> <pre>+CSSLCFG: 0,&lt;sslversion&gt;,&lt;authmode&gt;,&lt;ignoreltime&gt;,&lt;negotiatetime&gt;,&lt; ca_file&gt;,&lt;clientcert_file&gt;,&lt;clientkey_file&gt;  +CSSLCFG: 1,&lt;sslversion&gt;,&lt;authmode&gt;,&lt;ignoreltime&gt;,&lt;negotiatetime&gt;,&lt; ca_file&gt;,&lt;clientcert_file&gt;,&lt;clientkey_file&gt;  +CSSLCFG: 2,&lt;sslversion&gt;,&lt;authmode&gt;,&lt;ignoreltime&gt;,&lt;negotiatetime&gt;,&lt; ca_file&gt;,&lt;clientcert_file&gt;,&lt;clientkey_file&gt;  +CSSLCFG: 3,&lt;sslversion&gt;,&lt;authmode&gt;,&lt;ignoreltime&gt;,&lt;negotiatetime&gt;,&lt; ca_file&gt;,&lt;clientcert_file&gt;,&lt;clientkey_file&gt;  +CSSLCFG: 4,&lt;sslversion&gt;,&lt;authmode&gt;,&lt;ignoreltime&gt;,&lt;negotiatetime&gt;,&lt; ca_file&gt;,&lt;clientcert_file&gt;,&lt;clientkey_file&gt;  +CSSLCFG: 5,&lt;sslversion&gt;,&lt;authmode&gt;,&lt;ignoreltime&gt;,&lt;negotiatetime&gt;,&lt; ca_file&gt;,&lt;clientcert_file&gt;,&lt;clientkey_file&gt;</pre>

	<p>+CSSLCFG: 6,&lt;sslversion&gt;,&lt;authmode&gt;,&lt;ignoreltime&gt;,&lt;negotiatetime&gt;,&lt;ca_file&gt;,&lt;clientcert_file&gt;,&lt;clientkey_file&gt;</p> <p>+CSSLCFG: 7,&lt;sslversion&gt;,&lt;authmode&gt;,&lt;ignoreltime&gt;,&lt;negotiatetime&gt;,&lt;ca_file&gt;,&lt;clientcert_file&gt;,&lt;clientkey_file&gt;</p> <p>+CSSLCFG: 8,&lt;sslversion&gt;,&lt;authmode&gt;,&lt;ignoreltime&gt;,&lt;negotiatetime&gt;,&lt;ca_file&gt;,&lt;clientcert_file&gt;,&lt;clientkey_file&gt;</p> <p>+CSSLCFG: 9,&lt;sslversion&gt;,&lt;authmode&gt;,&lt;ignoreltime&gt;,&lt;negotiatetime&gt;,&lt;ca_file&gt;,&lt;clientcert_file&gt;,&lt;clientkey_file&gt;</p> <p><b>OK</b></p>
<p>Write Command /*Query the configuration of the specified SSL context*/ AT+CSSLCFG=&lt;ssl_ctx_index&gt;</p>	<p>Response +CSSLCFG: &lt;ssl_ctxindex&gt;,&lt;sslversion&gt;,&lt;authmode&gt;,&lt;ignoreltime&gt;,&lt;negotiatetime&gt;,&lt;ca_file&gt;,&lt;clientcert_file&gt;,&lt;clientkey_file&gt;</p> <p><b>OK</b></p>
<p>Write Command /*Configure the version of the specified SSL context*/ AT+CSSLCFG="sslversion",&lt;ssl_ctx_index&gt;,&lt;sslversion&gt;</p>	<p>Response a)If successfully: <b>OK</b> b)If failed: <b>ERROR</b></p>
<p>Write Command /*Configure the authentication mode of the specified SSL context*/ AT+CSSLCFG="authmode",&lt;ssl_ctx_index&gt;,&lt;authmode&gt;</p>	<p>Response a)If successfully: <b>OK</b> b)If failed: <b>ERROR</b></p>
<p>Write Command /*Configure the ignore local time flag of the specified SSL context*/ AT+CSSLCFG="ignorelocaltime",&lt;ssl_ctx_index&gt;,&lt;ignoreltime&gt;</p>	<p>Response a)If successfully: <b>OK</b> b)If failed: <b>ERROR</b></p>
<p>Write Command /*Configure the negotiate timeout value of the specified SSL context*/ AT+CSSLCFG="negotiatetime",&lt;ssl_ctx</p>	<p>Response a)If successfully: <b>OK</b> b)If failed:</p>

<b>&lt;_index&gt;,&lt;negotiatetime&gt;</b>	<b>ERROR</b>
Write Command /*Configure the server root CA of the specified SSL context*/ <b>AT+CSSLCFG="cacert",&lt;ssl_ctx_index&gt;,&lt;ca_file&gt;</b>	Response a)If successfully: <b>OK</b> b)If failed: <b>ERROR</b>
Write Command /*Configure the client certificate of the specified SSL context*/ <b>AT+CSSLCFG="clientcert",&lt;ssl_ctx_index&gt;,&lt;clientcert_file&gt;</b>	Response a)If successfully: <b>OK</b> b)If failed: <b>ERROR</b>
Write Command /*Configure the client key of the specified SSL context*/ <b>AT+CSSLCFG="clientkey",&lt;ssl_ctx_index&gt;,&lt;clientkey_file&gt;</b>	Response a)If successfully: <b>OK</b> b)If failed: <b>ERROR</b>

#### Defined Values

<b>&lt;ssl_ctx_index&gt;</b>	The SSL context ID. The range is 0-9.
<b>&lt;sslversion&gt;</b>	The SSL version, the default value is 4. 0 – SSL3.0 1 – TLS1.0 2 – TLS1.1 3 – TLS1.2 4 – All  The configured version should be support by server. So you should use the default value if you can't confirm the version which the server supported.
<b>&lt;authmode&gt;</b>	The authentication mode, the default value is 0. 0 – no authentication. 1 –server authentication. It needs the root CA of the server. 2 –server and client authentication. It needs the root CA of the server, the cert and key of the client. 3–client authentication and no server authentication. It needs the cert and key of the client.
<b>&lt;ignoretime&gt;</b>	The flag to indicate how to deal with expired certificate, the default value is 1. 0 – care about time check for certification. 1 – ignore time check for certification  When set the value to 0, it need to set the right current date and time by AT+CCLK when need SSL certification.

<b>&lt;negotiatetime&gt;</b>	<p>The timeout value used in SSL negotiate stage. The range is 10-300 seconds. The default value is 300.</p>
<b>&lt;ca_file&gt;</b>	<p>The root CA file name of SSL context. The file name must have type like “.pem” or “.der”.The length of filename is from 5 to 128 bytes.</p> <p>If the filename contains non-ASCII characters, the file path parameter should contain a prefix of {non-ascii} and the quotation mark (The string in the quotation mark should be hexadecimal of the filename’s UTF8 code).</p> <p>There are two ways to download certificate files to module:</p> <ol style="list-style-type: none"> <li>1. By AT+CCERTDOWN.</li> <li>2. By FTPS or HTTPS commands. Please refer to: SIM7500_SIM7600_SIM7800 Series_FTPS_AT Command Manual and SIM7500_SIM7600_SIM7800 Series_HTTP_AT Command Manual</li> </ol>
<b>&lt;clientcert_file&gt;</b>	<p>The client cert file name of SSL context. The file name must have type like “.pem” or “.der”.The length of filename is from 5 to 128 bytes.</p> <p>If the filename contains non-ASCII characters, the file path parameter should contain a prefix of {non-ascii} and the quotation mark (The string in the quotation mark should be hexadecimal of the filename’s UTF8 code).</p> <p>There are two ways to download certificate files to module:</p> <ol style="list-style-type: none"> <li>1. By AT+CCERTDOWN.</li> <li>2. By FTPS or HTTPS commands. Please refer to: SIM7500_SIM7600_SIM7800 Series_FTPS_AT Command Manual and SIM7500_SIM7600_SIM7800 Series_HTTP_AT Command Manual</li> </ol>
<b>&lt;clientkey_file&gt;</b>	<p>The client key file name of SSL context. The file name must have type like “.pem” or “.der”.The length of filename is from 5 to 128 bytes.</p> <p>If the filename contains non-ASCII characters, the file path parameter should contain a prefix of {non-ascii} and the quotation mark (The string in the quotation mark should be hexadecimal of the filename’s UTF8 code).</p> <p>There are two ways to download certificate files to module:</p> <ol style="list-style-type: none"> <li>1. By AT+CCERTDOWN.</li> <li>2. By FTPS or HTTPS commands. Please refer to: SIM7500_SIM7600_SIM7800 Series_FTPS_AT Command Manual and SIM7500_SIM7600_SIM7800 Series_HTTP_AT</li> </ol>

## 2.1.2 AT+CCERTDOWN Download certificate into the module

AT+CCERTDOWN Download certificate into the module	
Test Command AT+CCERTDOWN=?	Response +CCERTDOWN: (5-128),(1-10240)  <b>OK</b>
Write Command AT+CCERTDOWN=<filename>,<len>	Response a)If it can be download: > <input data here> b)If failed: <b>ERROR</b>

### Defined Values

<filename>	<p>The name of the certificate/key file. The file name must have type like “.pem” or “.der”. The length of filename is from 5 to 128 bytes.</p> <p>If the filename contains non-ASCII characters, the file path parameter should contain a prefix of {non-ascii} and the quotation mark (The string in the quotation mark should be hexadecimal of the filename’s UTF8 code).</p> <p>For example: If you want to download a file with name “中华.pem”, you should convert the “中华.pem” to UTF8 coding (&amp;#x4E2D;&amp;#x534E;.pem), then input the hexadecimal (262378344532443B262378353334453B2E70656D) of UTF8 coding.</p>
<len>	<p>The length of the file data to send. The range is from 1 to 10240 bytes.</p>

## 2.1.3 AT+CCERTLIST List certificates

AT+CCERTLIST List certificates	
Execute Command AT+CCERTLIST	Response [+CCERTLIST:<file_name> [+CCERTLIST:<file_name>] ...

	<CR><LF> <b>OK</b>
--	-----------------------

**Defined Values**

<filename>	The certificate/key files which has been downloaded to the module. If the filename contains non-ASCII characters, it will show the non-ASCII characters as UTF8 code.
------------	--

**2.1.4 AT+CCERTDELE Delete certificates**

**AT+CCERTDELE Delete certificate from the module**

Write Command <b>AT+CCERTDELE=&lt;filename&gt;</b>	Response a)If delete successfully: <b>OK</b> b)If failed: <b>ERROR</b>
---	--

**Defined Values**

<filename>	The name of the certificate/key file. The file name must have type like “.pem” or “.der”. The length of filename is from 5 to 128 bytes. If the filename contains non-ASCII characters, the file path parameter should contain a prefix of {non-ascii} and the quotation mark (The string in the quotation mark should be hexadecimal of the filename’s UTF8 code). For example: If you want to download a file with name “中华.pem”, you should convert the “中华.pem” to UTF8 coding (&#x4E2D;&#x534E;.pem), then input the hexadecimal (262378344532443B262378353334453B2E70656D) of UTF8 coding.
------------	--

**2.2 SSL Services AT**

**2.2.1 AT+CCHSET Configure the report mode of sending and receiving data**

AT+CCHSET is used to configure the mode of sending and receiving data. It must be called before AT+CCHSTART.

AT+CCHSET Configure the report mode of sending and receiving	
Test Command AT+CCHSET=?	Response +CCHSET: (0,1),(0,1)  <b>OK</b>
Read Command AT+CCHSET?	Response +CCHSET: <report_send_result>,<rcv_mode>  <b>OK</b>
Write Command AT+CCHSET=<report_send_result>[,<rcv_mode>]	Response a)If successfully: <b>OK</b> b)If failed: <b>ERROR</b>

#### Defined Values

<report_send_result>	Whether to report result of CCHSEND, the default value is 0: <u>0</u> – No. 1–Yes. Module will report +CCHSEND: <session_id>,<err> to MCU when complete sending data.
<rcv_mode>	The receiving mode: <u>0</u> – Output the data to MCU whenever received data. 1 – Module caches the received data and notifies MCU with +CCHEVENT: <session_id>, RECV EVENT. MCU can use AT+CCHRECV to receive the cached data (only in manual receiving mode).

### 2.2.2 AT+CCHMODE Configure the mode of sending and receiving data

AT+CCHMODE is used to elect transparent mode (data mode) or non-transparent mode (command mode). The default mode is non-transparent mode. This AT command must be called before calling AT+CCHSTART.

**NOTE:** There is only one session in the transparent mode, it's the first session.

AT+CCHMODE Configure the mode of sending and receiving	
Test Command AT+CCHMODE=?	Response +CCHMODE: (0,1)  <b>OK</b>
Read Command AT+CCHMODE?	Response +CCHMODE: <mode>

	<b>OK</b>
Write Command <b>AT+CCHMODE=&lt;mode&gt;</b>	Response a)If successfully: <b>OK</b> b)If failed: <b>ERROR</b>

#### Defined Values

<b>&lt;mode&gt;</b>	The mode value: <u>0</u> – Normal. 1 – Transparent mode The default value is 0.
---------------------	--

### 2.2.3 AT+CCHSTART Start SSL service

AT+CCHSTART is used to start SSL service by activating PDP context. You must execute AT+CCHSTART before any other SSL related operations.

<b>AT+CCHSTART Start SSL service</b>	
Execute Command <b>AT+CCHSTART</b>	Response a)If start SSL service successfully: <b>OK</b>  <b>+CCHSTART: 0</b> b)If start SSL service successfully: <b>+CCHSTART: 0</b>  <b>OK</b> c)If failed: <b>ERROR</b> d)If failed: <b>ERROR</b>  <b>+CCHSTART: &lt;err&gt;</b>
<b>Maximum Response Time</b>	120000ms

#### Defined Values

<b>&lt;err&gt;</b>	The result code, please refer to chapter 2.3.1
--------------------	--

## 2.2.4 AT+CCHSTOP Stop SSL service

AT+CCHSTOP is used to stop SSL service.

AT+CCHSTOP STOP SSL service	
Execute Command <b>AT+CCHSTOP</b>	Response a)If stop SSL service successfully: <b>+CCHSTOP: 0</b>  <b>OK</b> b)If stop SSL service successfully: <b>OK</b>  <b>+CCHSTOP: 0</b> c)If failed: <b>ERROR</b>

### Defined Values

<err>	The result code, please refer to chapter 2.3.1
-------	--

## 2.2.5 AT+CCHADDR Get the IPv4 address

AT+CCHADDR is used to get the IPv4 address after calling AT+CCHSTART.

AT+CCHSADDR Get the IPv4 address	
Execute Command <b>AT+CCHADDR</b>	Response <b>+CCHADDR: &lt; ip_address&gt;</b>  <b>OK</b>

### Defined Values

<ip_address>	A string parameter that identifies the IPv4 address after PDP activated.
--------------	--

## 2.2.6 AT+CCHSSLCFG Set the SSL context

AT+CCHSSLCFG is used to set the SSL context which to be used in the SSL connection. It must be called before AT+CCHOPEN and after AT+CCHSTART. The setting will be cleared after the CCHOPEN operation is finished.

**NOTE:** If you don't set the SSL context by this command before connecting to SSL/TLS server by AT+CCHOPEN, the CCHOPEN operation will use the SSL context as same as index <session\_id> (the 1st parameter of AT+CCHOPEN) when connecting to the server.

AT+CCHSSLCFG Set the SSL context	
Test Command AT+CCHSSLCFG=?	Response <b>+ CCHSSLCFG: (0,1),(0-9)</b>  <b>OK</b>
Read Command AT+CCHSSLCFG?	Response <b>+CCHSSLCFG: &lt;session_id&gt;,[&lt;ssl_ctx_index &gt;]</b> <b>+CCHSSLCFG: &lt;session_id&gt;,[&lt;ssl_ctx_index &gt;]</b>  <b>OK</b>
Write Command AT+CCHSSLCFG=<session_id>,<ssl_ctx_index>	Response a)If successfully: <b>OK</b> b)If failed: <b>ERROR</b>

#### Defined Values

<session_id>	The session_id to operate. It's from 0 to 1.
<ssl_ctx_index>	The SSL context ID which will be used in the SSL connection. Refer to the <ssl_ctx_index> of AT+CSSLCFG.

### 2.2.7 AT+CCHOPEN Connect to server

AT+CCHOPEN is used to connect to the server.

**NOTE:** If you don't set the SSL context by AT+CCHSSLCFG before connecting a SSL/TLS server by AT+CCHOPEN, it will use the <session\_id>( the 1'st parameter of AT+CCCHOPEN) SSL context when connecting to the server.

AT+CCHOPEN Connect to server	
Test Command AT+CCHOPEN=?	Response <b>+CCHOPEN: (0,1),"ADDRESS", (1-65535)[,(1-2)[,(1-65535)]]</b>  <b>OK</b>
Write Command AT+CCHOPEN=<session_id>, "<host>", <port>[<client_type>,[<bind_port>]]	Response a)If connect successfully: <b>+CCHOPEN: &lt;session_id&gt;,0</b>

	<p><b>OK</b></p> <p>b)If connect successfully:</p> <p><b>OK</b></p> <p>+CCHOPEN: &lt;session_id&gt;,0</p> <p>c)If connect successfully in transparent mode:</p> <p><b>CONNECT [&lt;text&gt;]</b></p> <p>d)If failed:</p> <p><b>OK</b></p> <p>+CCHOPEN: &lt;session_id&gt;,&lt;err&gt;</p> <p>e)If failed:</p> <p><b>ERROR</b></p> <p>f)If failed in transparent mode:</p> <p><b>CONNECT FAIL</b></p>
--	--

**Defined Values**

<session_id>	The session index to operate. It's from 0 to 1.
<host>	The server address, maximum length is 256 bytes.
<port>	The server port which to be connected, the range is from 1 to 65535.
<client_type>	The type of client: 1 - TCP client. 2 - SSL/TLS client.
<bind_port>	The local port for channel, the range is from 1 to 65535.
<text>	CONNECT result code string; the string formats please refer ATX/AT\V/AT&E command.
<err>	The result code: 0 is success. Other values are failure. Please refer to chapter 2.3.1

**2.2.8 AT+CCHCLOSE Disconnect from server**

AT+CCHCLOSE is used to disconnect from the server.

AT+CCHCLOSE Disconnect from the Server	
<p>Write Command</p> <p>AT+CCHCLOSE=&lt;session_id&gt;</p>	<p>Response</p> <p>a)If successfully:</p> <p>+CCHCLOSE: &lt;session_id&gt;,0</p> <p><b>OK</b></p>

	<p>b)If successfully: <b>OK</b></p> <p>+CCHCLOSE: &lt;session_id&gt;,0</p> <p>c)If successfully in transparent mode: <b>OK</b></p> <p><b>CLOSED</b></p> <p>d)If failed: <b>ERROR</b></p>
--	--

**Defined Values**

<session_id>	The session index to operate. It's from 0 to 1.
<err>	The result code: 0 is success. Other values are failure. Please refer to chapter 2.3.1.

**2.2.9 AT+CCHSEND Send data to server**

You can use AT+CCHSEND to send data to server.

AT+CCHSEND Send Data	
<p>Test Command AT+CCHSEND=?</p>	<p>Response:</p> <p>+CCHSEND: (0,1),(1-2048)</p> <p><b>OK</b></p>
<p>Read Command AT+CCHSEND?</p>	<p>Response:</p> <p>+CCHSEND: 0,&lt;unsent_len_0&gt;,1,&lt;unsent_len_1&gt;</p> <p><b>OK</b></p>
<p>Write Command AT+CCHSEND=&lt;session_id&gt;,&lt;len&gt;</p>	<p>Response</p> <p>a)if parameter is right:</p> <p>&gt;</p> <p>&lt;input data here&gt;</p> <p>When the total size of the inputted data reaches &lt;len&gt;, TA will report the following code. Otherwise, the serial port will be</p>

	<p>blocked.</p> <p><b>OK</b></p> <p>b)If parameter is wrong or other errors occur:</p> <p><b>ERROR</b></p>
--	--

**Defined Values**

<session_id>	The session_id to operate. It's from 0 to 1.
<len>	The length of data to send. Its range is from 1 to 2048 bytes.
<unsent_len_0>	The data of connection 0 cached in sending buffer which is waiting to be sent.
<unsent_len_1>	The data of connection 1 cached in sending buffer which is waiting to be sent.

**2.2.10 AT+CCHRECV Read the cached data that received from the server**

You can use AT+CCHRECV to read the cached data which received from the server.

AT+CCHRECV Receive the cached data that received from server	
<p>Read Command</p> <p>AT+CCHRECV?</p>	<p>Response</p> <p>+CCHRECV: LEN,&lt;cache_len_0&gt;,&lt;cache_len_1&gt;</p> <p><b>OK</b></p>
<p>Write Command</p> <p>AT+CCHRECV=&lt;session_id&gt;[,&lt;max_recv_len&gt;]</p>	<p>Response</p> <p>a)if parameter is right and there are cached data:</p> <p><b>OK</b></p> <p>[+CCHRECV: DATA, &lt;session_id&gt;,&lt;len&gt;</p> <p>...</p> <p>+CCHRECV: DATA, &lt;session_id&gt;,&lt;len&gt;</p> <p>...]</p> <p>+CCHRECV: &lt;session_id&gt;,&lt;err&gt;</p> <p>b) if parameter is not right or any other error occurs:</p> <p>+CCHRECV: &lt;session_id&gt;,&lt;err&gt;</p> <p><b>ERROR</b></p>

**Defined Values**

<session_id>	The session_id to operate. It's from 0 to 1.
<max_recv_len>	Maximum bytes of data to receive in the current AT+CCHRECV

	<p>calling. It will read all the received data when the value is greater than the length of RX data cached for session &lt;session_id&gt;.</p> <p>0 means the maximum bytes to receive is 2048 bytes. (But, when 2048 is greater than the length of RX data cached for session &lt;session_id&gt;, 0 means the length of RX data cached for session &lt;session_id&gt;).</p> <p>The default value is the length of RX data cached for session &lt;session_id&gt;.</p> <p><b>It will be not allowed when there is no data in the cache.</b></p>
<cache_len_0>	The length of RX data cached for connection 0.
<cache_len_1>	The length of RX data cached for connection 1.
<len>	The length of data followed.
<err>	The result code: 0 is success. Other values are failure. Please refer to chapter 2.3.1.

## 2.3 Command result codes and unsolicited codes

### 2.3.1 Command result <err> codes

Result codes	
0	Operation succeeded
1	Alerting state(reserved)
2	Unknown error
3	Busy
4	Peer closed
5	Operation timeout
6	Transfer failed
7	Memory error
8	Invalid parameter
9	Network error
10	Open session error
11	State error
12	Create socket error
13	Get DNS error
14	Connect socket error
15	Handshake error
16	Close socket error
17	Nonet
18	Send data timeout

19	Not set certificates
----	----------------------

### 2.3.2 Unsolicited result codes

Unsolicited codes	
+CCHEVENT: <session_id>,RECV EVENT	In manual receiving mode, when new data of a connection arriving to the module, this unsolicited result code will be reported to MCU.
+CCH_RECV_CLOSED: <session_id>,<err>	When receive data occurred any error, this unsolicited result code will be reported to MCU.
+CCH_PEER_CLOSED: <session_id>	The connection is closed by the server.

## 3 Example

Before all SSL related operations, we should ensure the following:

- a) ensure GPRS network is available:

**AT+CSQ**

**+CSQ: 23,0**

**OK**

**AT+CREG?**

**+CREG: 0,1**

**OK**

**AT+CGREG?**

**+CGREG: 0,1**

**OK**

- b) PDP context Enable:

```
// Specify the parameter value of the PDP context corresponding to cid
```

```
AT+CGSOCKCONT=1,"IP","CMNET"
```

```
OK
```

```
AT+CGPADDR
```

```
+CGPADDR: 1,10.49.14.68 //ensure the first PDP context get a IP address
```

```
+CGPADDR: 4,0.0.0.0
```

```
OK
```

**Note: usually CSOCKAUTH and CSOCKSETPN parameter are kept default if not care about.**

### 3.1 Access to TCP server

Following commands shows how to communicate with a TCP server.

```
// Enable reporting +CHSEND result
```

```
AT+CCHSET=1
```

```
OK
```

```
//start SSL service, activate PDP context
```

```
AT+CCHSTART
```

```
OK
```

```
+CCHSTART: 0
```

```
//connect to TCP server
```

```
AT+CCHOPEN=0,www.baidu.com,80,1
```

```
OK
```

```
+CCHOPEN: 0,0
```

```
//send data to server
```

```
AT+CCHSEND=0,121
```

```
>GET / HTTP/1.1
```

```
Host: www.baidu.com
```

```
User-Agent: Mozilla/5.0 (Windows NT 5.1; rv:2.0) Gecko/20100101 Firefox/4.0
```

**Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8**

**Accept-Language: zh-cn,zh;q=0.5**

**Accept-Encoding: gzip, deflate**

**Accept-Charset: GB2312,utf-8;q=0.7,\*;q=0.7**

**Keep-Alive: 115**

**Connection: keep-alive**

**Cookie: BAIDUID=D6F6D0D297CCAE39BD45C683996696C7:FG=1;**

**Hm\_lvt\_9f14aaa038bbba8b12ec2a4a3e51d254=1321597443439;**

**USERID=e194072f4759c0f7c2b6e5d3b09298984fd1**

**OK**

**+CCHSEND: 0,0**

//report the received data from server

**+CCHRECV: DATA,0,757**

**HTTP/1.1 302 Found**

**Connection: Keep-Alive**

**Content-Length: 225**

**Content-Type: text/html**

**Date: Wed, 05 Sep 2018 08:59:38 GMT**

**Location: https://www.baidu.com/**

**Server: BWS/1.1**

**Set-Cookie: BIDUPSID=D6F6D0D297CCAE39BD45C683996696C7; expires=Thu, 31-Dec-37 23:55:55 GMT; max-age=2147483647; path=/; domain=.baidu.com**

**Set-Cookie: PSTM=1536137978; expires=Thu, 31-Dec-37 23:55:55 GMT; max-age=2147483647; path=/; domain=.baidu.com**

**Set-Cookie: BD\_LAST\_QID=11878059346481009304; path=/; Max-Age=1**

**X-Ua-Compatible: IE=Edge,chrome=1**

**<html>**

**<head><title>302 Found</title></head>**

**<body bgcolor="white">**

**<center><h1>302 Found</h1></center>**

**<hr><center>7a367f7b87705e16b985e34ca59b8ae8b1d28d47**

**Time : Tue Aug 21 10:55:16 CST 2018</center>**

**</body>**

**</html>**

//Disconnect from the Service

**AT+CCHCLOSE=0**

**OK**

**+CCHCLOSE: 0**

//stop SSL Service

```
AT+CCHSTOP
```

```
OK
```

```
+CCHSTOP: 0
```

### 3.2 Access to SSL/TLS server (not verify server and client)

Following commands shows how to access to a SSL/TLS server without verifying the server. It needs to configure the authentication mode to 0, and then it will connect to the server successfully.

```
// Set the SSL version of the first SSL context
```

```
AT+CSSLCFG="sslversion",0,4
```

```
OK
```

```
// Set the authentication mode(not verify server) of the first SSL context
```

```
AT+CSSLCFG="authmode",0,0
```

```
OK
```

```
// Enable reporting +CHSEND result
```

```
AT+CCHSET=1
```

```
OK
```

```
// start SSL service, activate PDP context
```

```
AT+CCHSTART
```

```
OK
```

```
+CCHSTART: 0
```

```
// Set the first SSL context to be used in the SSL connection
```

```
AT+CCHSSLCFG=0,0
```

```
OK
```

```
//connect to SSL/TLS server
```

```
AT+CCHOPEN=0,"www.baidu.com", 443,2
```

```
OK
```

```
+CCHOPEN: 0,0
```

```
//send data to server
```

```
AT+CCHSEND=0,121
```

```
>GET / HTTP/1.1
```

```
Host: www.baidu.com
```

```
User-Agent: MAUI http User Agent
```

```
Proxy-Connection: keep-alive
```

Content-Length: 0

OK

+CCHSEND: 0,0

//report the received data from server

+CCHRECV: DATA,0,917

HTTP/1.1 200 OK

Accept-Ranges: bytes

Cache-Control: no-cache

Connection: Keep-Alive

Content-Length: 227

Content-Type: text/html

Date: Tue, 04 Sep 2018 06:21:35 GMT

Etag: "5b7b7f40-e3"

Last-Modified: Tue, 21 Aug 2018 02:56:00 GMT

P3p: CP=" OTI DSP COR IVA OUR IND COM "

Pragma: no-cache

Server: BWS/1.1

Set-Cookie: BD\_NOT\_HTTPS=1; path=/; Max-Age=300

Set-Cookie: BIDUPSID=D95046B2B3D5455BF01A622DB8DED9EA; expires=Thu, 31-Dec-37 23:55:55 GMT; max-age=2147483647; path=/; domain=.baidu.com

Set-Cookie: PSTM=1536042095; expires=Thu, 31-Dec-37 23:55:55 GMT; max-age=2147483647; path=/; domain=.baidu.com

Strict-Transport-Security: max-age=0

X-Ua-Compatible: IE=Edge,chrome=1

<html>

<head>

<script>

location.replace(location.href.replace("https://","http://"));

</script>

</head>

<body>

<noscript><meta http-equiv="refresh" content="0;url=http://www.baidu.com/"></noscript>

</body>

</html>

//Disconnect from the Service

AT+CCHCLOSE=0

OK

```
+CCHCLOSE: 0
```

```
//stop SSL Service
```

```
AT+CCHSTOP
```

```
OK
```

```
+CCHSTOP: 0
```

### 3.3 Access to SSL/TLS server (only verify the server)

Following commands shows how to access to a SSL/TLS server with verifying the server. It needs to configure the authentication mode to 1 and the right server root CA, and then it will connect to the server successfully.

```
// Set the SSL version of the first SSL context
```

```
AT+CSSLCFG="sslversion",0,4
```

```
OK
```

```
// Set the authentication mode(verify server) of the first SSL context
```

```
AT+CSSLCFG="authmode",0,1
```

```
OK
```

```
// Set the server root CA of the first SSL context
```

```
AT+CSSLCFG="cacert",0,"ca_cert.pem"
```

```
OK
```

```
// Enable reporting +CHSEND result
```

```
AT+CCHSET=1
```

```
OK
```

```
// start SSL service, activate PDP context
```

```
AT+CCHSTART
```

```
OK
```

```
+CCHSTART: 0
```

```
// Set the first SSL context to be used in the SSL connection
```

```
AT+CCHSSLCFG=0,0
```

```
OK
```

```
//connect to SSL/TLS server
```

```
AT+CCHOPEN=0,"www.baidu.com",443,2
```

```
OK
```

+CCHOPEN: 0,0

//send data to server

**AT+CCHSEND=0,121**

>GET / HTTP/1.1

Host: www.baidu.com

User-Agent: MAUI http User Agent

Proxy-Connection: keep-alive

Content-Length: 0

OK

+CCHSEND: 0,0

//report the received data from server

+CCHRECV: DATA,0,917

HTTP/1.1 200 OK

Accept-Ranges: bytes

Cache-Control: no-cache

Connection: Keep-Alive

Content-Length: 227

Content-Type: text/html

Date: Tue, 04 Sep 2018 06:21:35 GMT

Etag: "5b7b7f40-e3"

Last-Modified: Tue, 21 Aug 2018 02:56:00 GMT

P3p: CP=" OTI DSP COR IVA OUR IND COM "

Pragma: no-cache

Server: BWS/1.1

Set-Cookie: BD\_NOT\_HTTPS=1; path=/; Max-Age=300

Set-Cookie: BIDUPSID=D95046B2B3D5455BF01A622DB8DED9EA; expires=Thu, 31-Dec-37 23:55:55 GMT; max-age=2147483647; path=/; domain=.baidu.com

Set-Cookie: PSTM=1536042095; expires=Thu, 31-Dec-37 23:55:55 GMT; max-age=2147483647; path=/; domain=.baidu.com

Strict-Transport-Security: max-age=0

X-Ua-Compatible: IE=Edge,chrome=1

<html>

<head>

<script>

location.replace(location.href.replace("https://","http://"));

</script>

</head>

<body>

<noscript><meta http-equiv="refresh" content="0;url=http://www.baidu.com/"></noscript>

```
</body>
</html>
//Disconnect from the Service
AT+CCHCLOSE=0
OK

+CCHCLOSE: 0
//stop SSL Service
AT+CCHSTOP
OK

+CCHSTOP: 0
```

### 3.4 Access to SSL/TLS server (verify server and client)

Following commands shows how to access to a SSL/TLS server with verifying the server and client. It needs to configure the authentication mode to 2, the right server root CA, the right client certificate and key, and then it will connect to the server successfully.

```
// Set the SSL version of the first SSL context
AT+CSSLCFG="sslversion",0,4
OK

// Set the authentication mode(verify server and client) of the first SSL context
AT+CSSLCFG="authmode",0,2
OK

// Set the server root CA of the first SSL context
AT+CSSLCFG="cacert",0,"ca_cert.pem"
OK

// Set the client certificate of the first SSL context
AT+CSSLCFG="clientcert",0,"cert.pem"
OK

// Set the client key of the first SSL context
AT+CSSLCFG="clientkey",0,"key_cert.pem"
```

**OK**

// Enable reporting +CHSEND result

**AT+CCHSET=1**

**OK**

// start SSL service, activate PDP context

**AT+CCHSTART**

**OK**

**+CCHSTART: 0**

// Set the first SSL context to be used in the SSL connection

**AT+CCHSSLCFG=0,0**

**OK**

//connect to SSL/TLS server

**AT+CCHOPEN=0, "www.baidu.com",443,2**

**OK**

**+CCHOPEN: 0,0**

//send data to server

**AT+CCHSEND=0,121**

>GET / HTTP/1.1

Host: www.baidu.com

User-Agent: MAUI http User Agent

Proxy-Connection: keep-alive

Content-Length: 0

**OK**

**+CCHSEND: 0,0**

//report the received data from server

**+CCHRECV: DATA,0,917**

**HTTP/1.1 200 OK**

**Accept-Ranges: bytes**

**Cache-Control: no-cache**

**Connection: Keep-Alive**

**Content-Length: 227**

**Content-Type: text/html**

**Date: Tue, 04 Sep 2018 06:21:35 GMT**

**Etag: "5b7b7f40-e3"**

**Last-Modified: Tue, 21 Aug 2018 02:56:00 GMT**

**P3p: CP=" OTI DSP COR IVA OUR IND COM "**

**Pragma: no-cache**

```
Server: BWS/1.1
Set-Cookie: BD_NOT_HTTPS=1; path=/; Max-Age=300
Set-Cookie: BIDUPSID=D95046B2B3D5455BF01A622DB8DED9EA; expires=Thu, 31-Dec-37 23:55:55
GMT; max-age=2147483647; path=/; domain=.baidu.com
Set-Cookie: PSTM=1536042095; expires=Thu, 31-Dec-37 23:55:55 GMT; max-age=2147483647; path=/;
domain=.baidu.com
Strict-Transport-Security: max-age=0
X-Ua-Compatible: IE=Edge,chrome=1
```

```
<html>
<head>
  <script>
    location.replace(location.href.replace("https://","http://"));
  </script>
</head>
<body>
  <noscript><meta http-equiv="refresh" content="0;url=http://www.baidu.com/"></noscript>
</body>
</html>
```

```
//Disconnect from the Service
```

```
AT+CCHCLOSE=0
```

```
OK
```

```
+CCHCLOSE: 0
```

```
//stop SSL Service
```

```
AT+CCHSTOP
```

```
OK
```

```
+CCHSTOP: 0
```

### 3.5 Access to SSL/TLS server (only verify the client)

Following commands shows how to access to a SSL/TLS server with verifying the client. It needs to configure the authentication mode to 3, the right client certificate and key, and then it will connect to the server successfully.

```
// Set the SSL version of the first SSL context
```

```
AT+CSSLCFG="sslversion",0,4
```

```
OK
```

```
// Set the authentication mode(only verify client) of the first SSL context
```

```
AT+CSSLCFG="authmode",0,3
```

```
OK
```

```
// Set the client certificate of the first SSL context
```

```
AT+CSSLCFG="clientcert",0,"cert.pem"
```

```
OK
```

```
// Set the client key of the first SSL context
```

```
AT+CSSLCFG="clientkey",0,"key_cert.pem"
```

```
OK
```

```
// Enable reporting +CHSEND result
```

```
AT+CCHSET=1
```

```
OK
```

```
// start SSL service, activate PDP context
```

```
AT+CCHSTART
```

```
OK
```

```
+CCHSTART: 0
```

```
// Set the first SSL context to be used in the SSL connection
```

```
AT+CCHSSLCFG=0,0
```

```
OK
```

```
//connect to SSL/TLS server
```

```
AT+CCHOPEN=0, "www.baidu.com", 443,2
```

```
OK
```

```
+CCHOPEN: 0,0
```

```
//send data to server
```

```
AT+CCHSEND=0,121
```

```
>GET / HTTP/1.1
```

```
Host: www.baidu.com
```

```
User-Agent: MAUI http User Agent
```

```
Proxy-Connection: keep-alive
```

```
Content-Length: 0
```

```
OK
```

```
+CCHSEND: 0,0
```

```
//report the received data from server
```

```
+CCHRECV: DATA,0,917
HTTP/1.1 200 OK
Accept-Ranges: bytes
Cache-Control: no-cache
Connection: Keep-Alive
Content-Length: 227
Content-Type: text/html
Date: Tue, 04 Sep 2018 06:21:35 GMT
Etag: "5b7b7f40-e3"
Last-Modified: Tue, 21 Aug 2018 02:56:00 GMT
P3p: CP=" OTI DSP COR IVA OUR IND COM "
Pragma: no-cache
Server: BWS/1.1
Set-Cookie: BD_NOT_HTTPS=1; path=/; Max-Age=300
Set-Cookie: BIDUPSID=D95046B2B3D5455BF01A622DB8DED9EA; expires=Thu, 31-Dec-37 23:55:55
GMT; max-age=2147483647; path=/; domain=.baidu.com
Set-Cookie: PSTM=1536042095; expires=Thu, 31-Dec-37 23:55:55 GMT; max-age=2147483647; path=/;
domain=.baidu.com
Strict-Transport-Security: max-age=0
X-Ua-Compatible: IE=Edge,chrome=1

<html>
<head>
  <script>
    location.replace(location.href.replace("https://","http://"));
  </script>
</head>
<body>
  <noscript><meta http-equiv="refresh" content="0;url=http://www.baidu.com/"></noscript>
</body>
</html>

//Disconnect from the Service
AT+CCHCLOSE=0
OK

+CCHCLOSE: 0
//stop SSL Service
AT+CCHSTOP
OK

+CCHSTOP: 0
```

### 3.6 Access to SSL/TLS server in transparent mode

Following commands shows how to access to a SSL/TLS server with not verifying the server in transparent mode. It needs to configure the sending and receiving mode to 1(the transparent mode). Only the session 0 is support the transparent mode.

```
// Set the transparent mode
AT+CCHMODE=1
OK
// Enable reporting +CHSEND result
AT+CCHSET=1
OK
// start SSL service, activate PDP context
AT+CCHSTART
OK

+CCHSTART: 0
// Set the first SSL context to be used in the SSL connection
AT+CCHSSLCFG=0,0
OK
//connect to SSL/TLS server
AT+CCHOPEN=0, "www.baidu.com", 443,2
CONNECT 115200
//send data to server
GET / HTTP/1.1
Host: www.baidu.com
User-Agent: MAUI http User Agent
Proxy-Connection: keep-alive
Content-Length: 0

//report the received data from server
HTTP/1.1 200 OK
Accept-Ranges: bytes
Cache-Control: no-cache
Connection: Keep-Alive
Content-Length: 227
Content-Type: text/html
Date: Tue, 04 Sep 2018 06:26:03 GMT
Etag: "5b7b7f40-e3"
```

```
Last-Modified: Tue, 21 Aug 2018 02:56:00 GMT
P3p: CP=" OTI DSP COR IVA OUR IND COM "
Pragma: no-cache
Server: BWS/1.1
Set-Cookie: BD_NOT_HTTPS=1; path=/; Max-Age=300
Set-Cookie: BIDUPSID=F19D0F1E532ED84CE275BC1006F91F9E; expires=Thu, 31-Dec-37 23:55:55
GMT; max-age=2147483647; path=/; domain=.baidu.com
Set-Cookie: PSTM=1536042363; expires=Thu, 31-Dec-37 23:55:55 GMT; max-age=2147483647; path=/;
domain=.baidu.com
Strict-Transport-Security: max-age=0
X-Ua-Compatible: IE=Edge,chrome=1
```

```
<html>
<head>
  <script>
    location.replace(location.href.replace("https://","http://"));
  </script>
</head>
<body>
  <noscript><meta http-equiv="refresh" content="0;url=http://www.baidu.com/"></noscript>
</body>
</html>
```

```
//switch to command mode
```

```
+++
```

```
OK
```

```
//Disconnect from the Service
```

```
AT+CCHCLOSE=0
```

```
OK
```

```
CLOSED
```

```
//stop SSL Service
```

```
AT+CCHSTOP
```

```
OK
```

```
+CCHSTOP: 0
```

### 3.7 Download certificate into module

Following commands shows how to download certificate into module.

```
// download file with ASCII coding file name
```

```
AT+CCERTDOWN="client_key.der",1702
```

```
> -----BEGIN RSA PRIVATE KEY-----
```

```
MIIEowIBAAKCAQEAlwuz/TNa+foGBG6rXpWE1Wnuc+GN9vS7MRENKOH+z2UfGuaV
```

```
BSb8VYFCgoL4RnWLwXAcLlaqw88zLCN89EK6IydaAwNmI/U6nu3oPsVkn8r9+sOX
```

```
yh9VD01DmSU349QWJvRgt1ocsFI1VTdd6RDkVtu7FdKv4XC5WHcOD7yrEIsVa7+G
```

```
Qbnm5cCCz8E75HH8vHZAOfEaV3HvIHnh/1RZ+jh4ysyhEmFNOFCn3r9v2yu4kPRX
```

```
43xESB13Ue4HgSbnT+Q7LIEK+dfsmUBoSpsS2NAmqOiqGrmmYygT3/V/ISX54hit
```

```
gli5bvg9DuNHYBwh2C+4nyZF95pMj2dEJf4jNwIDAQABAoIBAAJ9ze06QKDo79p4
```

```
3NjFjJhck/NTYB0XsIk/+iDhgWt4VogCD6kzGGxsomU2tdOrsq9xIvXcthpEU5IQ
```

```
98mrpBhaWNC96JxlOh9O+0q1xNAh8AiH22QZGjUTaC8Jfx+B6w+fbkz37os1/+00
```

```
6ZajkbChFTfp7r7ANj5wUEoQKZ4vNpLJxLWDk6uH4ZMNveWcBaZQ21TUg9ZmoskK
```

```
EJ2ZEr/3kOSBgi2B6F50zyL8f1mbqPahHNLqtrndV5/Lr4n74TqZXRwt5CI9GrBv
```

```
tYXDhc+5Y7e1TUIXV00AMDik+3cVR8m8Oa20tSdXjcw2iUk9brxb4uxreOouGfPW
```

```
5IO+q1ECgYEA4Kkok17DVx5FiapFQvJ2Jqi2/WhzDncuBGbZtcLZnwRVfkPn3cBZ
```

```
JGNwxYyfEdwItPvTYQYh6Qg81XRdSRfF43GzkQXNmkPOdZM0x3tFwzV6K5Fg7aeR
```

```
g50UddaA9MraClOgK++7C6BvA3ImXciK4VWeSZOmDW99Y6mgf92RdkCgYEArb2u
```

```
/Id72LGQBmx0Z+36Hf1dxo6RQ+dB+m6XBMR8iuB/jGO/5PHdFoKoF2qa9Yj2W1+X
```

```
B29Xmc1HS6GTvkDI5N5JXNO7fDmlAxd5whbwDdemv3VEt8xJ2UeACIawjKtVcFoH
```

```
LRNlvDBttWVvICZg+9HfVpuPm14oFxn/HtSxt48CgYACxDJ6thUDspy6mD0oGOI5
```

```
kaRHNI0OJYuMhFOz+EVDvwLqfh2RzneKiiruU8/1oVb+G4e7zx6FxxMwsbEgYEmQ
```

```
hmrmo0Kn3qPhMMHanvr572Oku7KM2p5hF4MT/GM0IHdU31D1JrTcJap1TVomAaCL
```

```
FqY88arQFwFSz8HfIe0r6QKBgCbQLtTdzKzqJdt8+6cwQFYg+9O59MJGVVefNskp
```

```
chhzVfAX0n9TI5Lq9fMJ5FX4g+3JGargjfWuGCTTFBk0TM2t4wde7AmwiiiU5LU
```

```
T2Afo6pLTKrSE9k+yX2iug+O156VfsbIeAm/Ng5RCJ91JCvFgULro6/axNmnWORF
```

```
9rK7AoGBAIK4edrX1MjercsLu3y9Dy4pAx6ER6ei4xpko25U8wUcqqc+YD2m2x1A
```

```
DjqROIteaxXkmPIyRKAXVarhk8LmXT/oDFUAPsTqUZ9LBrviqtMi+G2OFPbdKDwe
```

```
ZBNAgwFpFIUVoi0UYnZF8rBq0tepqivrayEWdKKfMMJjq+172SxD
```

-----END RSA PRIVATE KEY-----

OK

// download file with not ASCII coding file name

[AT+CCERTDOWN={non-ascii}"262378344532443B262378353334453B2E70656D",1918](#)

>-----BEGIN CERTIFICATE-----

MIIFRDCC AyygAwIBAgII ZmPau7FelQswDQYJKoZIhvcNAQELBQA wQDELMAkGA1UE  
BhMCU0kxGzAZBgNVBAoME nN0YXRILWluc3RpdHV0aW9uczEUMBIGA1UEAwwLVGF4  
IENBIFRlc3QwHhcNMTUwNzIzMTUyOTA1WhcNMzUwNzIzMTUyOTA1WjBAMQswCQYD  
VQQGEwJTSTEBMBkGA1UECgwSc3Rh dGUtaW5zdG10dXRpb25zMRQwEgYDVQQDDAtU  
YXggQ0EgVGVzdDCCAiIwDQYJKoZIhvcNAQEBBQADggIPADCC AgoCggIBALmH3XNA  
KDgN8+G2jX4W/a7LTER10VbRhkGeuc9zyOuj9gigYXLno4Im/S4iXMcCs1IxsSj  
NJ1YMOje4qgHbFKQwWV588VDw7/fiMMZIXvFjHfladdHASEDMT53bKX3HI dJZ/iL  
6xhpJ/+C/I8dnWcMZUkeP+9BUAni/I2xrHaAVIli0aS6uc/DjO7b4Gj1VI4FGIH o  
DIH+LmWz26P2gg2xnpWgIxXzs5sN8nYErwu+6h/9xREHco8PPCAZb5HZhqoIzYzk  
N1S1Do6qAzt/wJM0mhWOWHt9fhp/RoYQ5ZFCIZmgd1cJcr6S6U7ebAQ+yYRsIWU5  
+FLYZ4Zlt3ZAHNWyraMee/kFsaGcO21cwE+tPDOIn41B8XvfaXApQt4+TejZWzoH  
V0ojA+9H8V+wCFVMJssViFOzuS6SIEZ/xz slo+B//cfUkq/PnWLJHEy4BJXsj4+F  
CvliZ7Lq3B/RcQmBjmTRQ0mxahiMGrrQW4TLjUYgY8IfwKfMfwFwVwUyk5br9Gr s  
UX7jy7+Xx17Qed4p0jjOC7KutzRIGr6ULSk11qpd5IHeIwzSOaTXk6rAzZYupPH5  
KvY65mdRfq0C0cB2bMvk9m9lyeLfZz5+L9XDLlodTdwOeWaKvjFErT8WSEkphXtG  
q13TVgicoxsHC2K+8hpFjpaz69ZCmTzj4/17AgMBAAGjQjBAMB0GA1UdDgQWB BQz  
zVr7CUfHAeY2KCb1gXy3jjX3sjAPBgNVHRMBAf8EBTADAQH/MA4GA1UdDwEB/wQE  
AwIBBjANBgkqhkiG9w0BAQsFAAOC AgEAR9xtbaNa/jSAAyqe3aq88GG7rCyxROGH  
BPcakfMmhx1cLYdcY5ATXL/n67eo+S+1g7e/sK3fVXav5qWs9oUEhAOgcOAC Mohu  
JIBbMq2Qp8lxdpiRWCcyiY1vGQcHcZ02oey/c06fBZE4iqJdYAhYhsBB5H+idtwJ  
s6Lade4wqG58hWCNKbXU+KWDckGGX5CxsFu7gdYgyKq0ow60qQWi4H8pD+W01Bn  
rvIskAT7vMk2BOz+YICKZmuq0h3PCKk5T6xA01fUZCaeze0RozFaekDBEHK0bc1D  
My3SKbB3cjdcMzmV8sVdxnNOTxlrP7+BinctxT3q3Va96kTmwI5pD0x6KOWC7Urr

```
53ubhI3U2XBazkk14IDLU+7tqBqhDWwIMN0NyW1MRTF8JB9Rz+4yCcDWMOT/FZg7
C60RrcnaO/0GETDz6XI6zedBXo1Q/rJTtXMOr8iVnc+joZyO2ImOuTwP3C7M3Bnp
gFHqDtD48n9PV9prhbD4fYPyMe/3rshtBcpGAy2cGjpsP28pkvP8lwBaP8pnpvxQ
7d3oiCBzznaOHjhm8+8C53b/1txzj/LP/4ZzIynsOhxy4cihEPhAg1MKUY9qnbw9
9Q6EKrCSqk3TPqiWrTtu4pxyiEiquCHk8n+HX5cVhxUkaEShdx4bjgvKB7JRF2T2
ST1lrKEM2DY=
```

-----END CERTIFICATE-----

OK

//list certificate files

**AT+CCERTLIST**

+CCERTLIST: "&#x4E2D;&#x534E;.pem"

+CCERTLIST: "client\_key.der"

OK