GSM HTTPS

Application Note

GSM/GPRS Module Series

Rev. GSM_HTTPS_Application_Note_V3.3

Date: 2020-01-13

Status: Released

Our aim is to provide customers with timely and comprehensive service. For any assistance, please contact our company headquarters:

Quectel Wireless Solutions Co., Ltd.
Building 5, Shanghai Business Park Phase III (Area B), No.1016 Tianlin Road, Minhang District, Shanghai, China 200233
Tel: +86 21 5108 6236
Email: info@quectel.com

Or our local office. For more information, please visit:
http://www.quectel.com/support/sales.htm

For technical support, or to report documentation errors, please visit:
http://www.quectel.com/support/technical.htm
Or email to: support@quectel.com

GENERAL NOTES

COPYRIGHT

About the Document

History

Revision	Date	Author	Description
3.0	2015-12-10	Oven TAO	Initial
3.1	2016-12-23	Oven TAO	1. Updated AT+QSSLCFG command in Chapter 2.2.1 2. Modified the example in Chapter 3.3
3.2	2017-01-22	Sandy YE	1. Updated AT+QSSLCFG command in Chapter 2.2.1 2. Added examples in Chapter 3.3.1 and 3.3.2
3.3	2020-01-13	Jaryoung LI	Updated the description of the server root CA certificate in Chapter 2.2, 3.1, 3.2 and 3.3.

Contents

Table Index

1 Introduction

This document mainly introduces how to use the HTTPS function of Quectel GSM modules. HTTPS is used to secure the data transmission.

Hypertext Transfer Protocol Secure (HTTPS) is a combination of the Hypertext Transfer Protocol (HTTP) with SSL/TLS protocols to provide encrypted communication and secure identification of a network web server. HTTPS is the result of simply layering the HTTP on the top of the SSL/TLS protocols, thus adding the security capabilities of SSL/TLS to standard HTTP communication.

In some cases, in order to ensure communication privacy, the communication between the server and the client should be in an encrypted way, and SSL function can prevent data from being eavesdropped, tampered, or forged during the communication process.

This document is applicable to Quectel GSM modules.

1.1. SSL Version and Cipher Suite

The following SSL versions are supported by Quectel GSM modules currently.

Table 1: Supported SSL Versions

Supported SSL Versions
SSL3.0
TLS1.0
TLS1.1
TLS1.2

The following table shows the SSL cipher suites supported by Quectel GSM modules. For detailed description of cipher suites, please refer to *RFC 2246-The TLS Protocol Version 1.0*.

Table 2: Supported SSL Cipher Suites

Supported SSL Cipher Suites	
0X0035	TLS_RSA_WITH_AES_256_CBC_SHA
0X0005	TLS_RSA_WITH_RC4_128_SHA
0X0004	TLS_RSA_WITH_RC4_128_MD5
0X000A	TLS_RSA_WITH_3DES_EDE_CBC_SHA
0X002F	TLS_RSA_WITH_AES_128_CBC_SHA
0X003D	TLS_RSA_WITH_AES_256_CBC_SHA256

1.2. The Procedure of Using SSL Function

Step 1: Install certificate and key to RAM or NVRAM by **AT+QSECWRITE** command. **AT+QSECDEL** is used to delete the certificate and key, and **AT+QSECREAD** is used to check the checksum of certificate and key. If the server and client authentication is not needed, please skip this step.

Step 2: Configure the APN, username, password of context by **AT+QICSGP** command. **AT+QIREGAPP** is used to register on TCP/IP stack.

Step 3: Activate GPRS PDP context by **AT+QIACT** command. After the PDP context has been activated, please query the local IP address by **AT+QILOCIP** command.

Step 4: Configure SSL version, cipher suit, server authentication, client authentication, server root CA certificate, client certificate and client key by **AT+QSSLCFG** command.

Step 5: Configure URL by **AT+QHTTPURL** command. After **CONNECT** is returned, enter URL in the format of: "https:URL".

Step 6: Send HTTP GET request by **AT+QHTTPGET** command.

Step 7: Read HTTP server response by **AT+QHTTPREAD** command.

1.3. Error Handling

1.3.1. PDP Activation Fails

If PDP context is failed to be activated by **AT+QIACT** command, please check the following configurations:

1. Query whether the PS domain is attached by **AT+CGATT?** command. If not, execute **AT+CGATT=1** command to attach PS domain.

2. Query **AT+CGREG** status by **AT+CGREG?** command and make sure the PS domain is registered.
3. Query the PDP context parameters by **AT+QIREGAPP** command and make sure the APN of specified PDP context is set.
4. Make sure the specified PDP context ID is neither used by PPP nor activated by **AT+CGACT** command.
5. The module only supports two PDP contexts activated simultaneously, so please make sure the amount of activated PDP context is no more than 2.

If all above configurations are confirmed, but the result of executing command **AT+QIACT** always fails, please reboot the module to resolve this issue. After rebooting the module, please check the configurations mentioned above at least three times at an interval of 10 minutes to avoid frequent rebooting of the module.

2 Description of AT Commands

2.1. AT Command Syntax

Table 3: Types of AT Commands and Responses

Test Command	**AT+<cmd>=?**	This command returns the list of parameters and value ranges set by the corresponding Write Command or internal processes.
Read Command	**AT+<cmd>?**	This command returns the currently set value of the parameter or parameters.
Write Command	**AT+<cmd>=<p1>[,<p2>[,<p3>[...]]]**	This command sets the user-definable parameter values.
Execution Command	**AT+<cmd>**	This command reads non-variable parameters affected by internal processes in the module.

NOTES

1. **<…>**: Parameter name. Angle brackets do not appear on the command line. The parameter value indicated by "_" is the default one.
2. **[…]**: Optional parameter. Square brackets do not appear on the command line. When an optional parameter is omitted, the default value will be used unless otherwise specified.

2.2. Description of AT Commands

2.2.1. AT+QSSLCFG SSL Configuration

This AT command is used to configure the SSL version, cipher suite, secure level, server root CA certificate, client certificate, client key, RTC time ignorance and SSL context index of HTTP/HTTPS. These parameters will be used in the handshake procedure.

CTX is the abbreviation of SSL context. **<CTX_index>** is the index of the SSL context. Quectel GSM modules support six SSL contexts at most. And on the basis of an SSL context, several SSL connections

can be established. The settings such as SSL version and cipher suite are stored in the SSL context, and they will be applied to a new SSL connection which is associated with the SSL context.

AT+QSSLCFG SSL Configuration	
Test Command **AT+QSSLCFG=?**	Response **+QSSLCFG: "type",(**list of supported **<CTX_index>**s**),"value"** **OK**
Read Command Query settings of the context **AT+QSSLCFG="ctxindex",<CTX_index>**	Response **+QSSLCFG: <CTX_index>,<SSL_version>,<seclevel>,<cipher_suite>,<cacert>,<client_cert_name>,<client_key_name>** **OK** Or **ERROR**
Write Command Configure SSL version **AT+QSSLCFG="sslversion",<CTX_index>[,<SSL_version>]**	Response If **<SSL_version>** is omitted, query the SSL version: **+QSSLCFG: "sslversion",<SSL_version>** **OK** If **<SSL_version>** is specified, set the SSL version: **OK** Or **ERROR**
Write Command Configure cipher suite **AT+QSSLCFG="ciphersuite",<CTX_index>[,<cipher_suite>]**	Response If **<cipher_suites>** is omitted, query the cipher suites: **+QSSLCFG: "ciphersuite",<cipher_suite>** **OK** If the **<**list of supported **<cipher_suites>**s**>** is specified, set the cipher suite: **OK** Or **ERROR**
Write Command Configure authentication mode **AT+QSSLCFG="seclevel",<CTX_index>[,<seclevel>]**	Response If **<sec_level>** is omitted, query the authentication mode: **+QSSLCFG: "seclevel",<seclevel>** **OK**

	If **<seclevel>** is specified, set the authentication mode: **OK** Or **ERROR**
Write Command Configure the server root CA certificate **AT+QSSLCFG="cacert",<CTX_index>[,<CA_cert_name>]**	Response If **<CA_cert_name>** is omitted, query the path of server root CA certificate: **+QSSLCFG: "cacert",<CA_cert_name>** **OK** If **<CA_cert_name>** is specified, set the path of server root CA certificate: **OK** Or **ERROR**
Write Command Configure the client certificate **AT+QSSLCFG="clientcert",<CTX_index>[,<client_cert_name>]**	Response If **<client_cert_name>** is omitted, query the client certificate: **+QSSLCFG: "clientcert",<client_cert_name>** **OK** If **<client_cert_name>** is specified, set the client certificate: **OK** Or **ERROR**
Write Command Configure the client key **AT+QSSLCFG="clientkey",<CTX_index>[,<client_key_name>]**	Response If **<client_key_name>** is omitted, query the path of client key: **+QSSLCFG: "clientkey",<client_key_name>** **OK** If **<client_key_name>** is specified, set the path of client key: **OK** Or **ERROR**
Write Command Configure whether to ignore the RTC time **AT+QSSLCFG="ignorertctime"[,<ignore_RTC_time>]**	Response If **<ignore_RTC_time>** is omitted, query whether the RTC time is ignored: **+QSSLCFG: "ignorertctime",<ignore_RTC_time>** **OK**

	If **<ignore_RTC_time>** is specified, set whether to ignore the RTC time: **OK** Or **ERROR**
Write Command Enable/Disable the HTTPS function **AT+QSSLCFG="https"[,<HTTPS_enable>]**	Response If **<HTTPS_enable>** is omitted, query whether to enable HTTPS function: **+QSSLCFG: "https",<HTTPS_enable>** **OK** If **<HTTPS_enable>** is specified, set whether to enable HTTPS function: **OK** Or **ERROR**
Write Command Configure SSL context index for HTTPS **AT+QSSLCFG="httpsctxi"[,<HTTPS_CTX_index>]**	Response If **<HTTPS_CTX_index>** is omitted, query the SSL context index for HTTPS: **+QSSLCFG: "httpsctxi",<HTTPS_CTX_index>** **OK** If **<HTTPS_CTX_index>** is specified, set the SSL context for HTTPS: **OK** Or **ERROR**
Maximum Response Time	300ms
Characteristics	The command takes effect immediately. The configurations will not be saved.

Parameter

<CTX_index>	Integer type. SSL context index. Range: 0-5.
<SSL_version>	Integer type. Configure the supported SSL version. 0 SSL3.0 1 TLS1.0 2 TLS1.1 3 TLS1.2 4 All supported
<cipher_suite>	Configure the cipher suite.

	0X0035 TLS_RSA_WITH_AES_256_CBC_SHA 0X002F TLS_RSA_WITH_AES_128_CBC_SHA 0X0005 TLS_RSA_WITH_RC4_128_SHA 0X0004 TLS_RSA_WITH_RC4_128_MD5 0X000A TLS_RSA_WITH_3DES_EDE_CBC_SHA 0X003D TLS_RSA_WITH_AES_256_CBC_SHA256
<seclevel>	Integer type. Configure the authentication mode. 0 No authentication 1 Manage server authentication 2 Manage server and client authentication if requested by the remote server
<CA_cert_name>	String format. Configure the server root CA certificate.
<client_cert_name>	String format. Configure the client certificate.
<client_key_name>	String format. Configure the client key.
<ignore_RTC_time>	Integer type. Configure whether to ignore the RTC time. 0 Do not ignore the RTC time 1 Ignore the RTC time
<HTTPS_enable>	Integer type. Enable/disable the HTTPS function. 0 Disable HTTPS 1 Enable HTTPS
<HTTPS_CTX_index>	Integer type. SSL context for HTTPS. It is the index of SSL context. Range: 0-5. If the host does not configure it, the value is -1.

NOTES

1. The format of **<CA_cert_name>**, **<client_cert_name>** and **<client_key_name>** can be as follows:
   "RAM:filename" File is uploaded to RAM
   "NVRAM:filename" File is uploaded to NVRAM. Server root CA certificate, one client certificate and one client private key are supported. The filename of server root CA certificate must be *CA0*, the filename of client certificate must be *CC0*, and the filename of client private key must be *CK0*.
   CA0 Identify a server root CA certificate
   CC0 Identify a client certificate
   CK0 Identify a client private key
2. If no authentication is set, security data will not be needed. If server authentication has been set, server root CA certificate needs to be configured. If both server and client authentications have been set, the client certificate, server root CA certificate and client private key need to be configured.

2.2.2. AT+QSECWRITE Add a Certificate or Key

This command is used to add user certificate, user key and server root CA certificate to RAM or NVRAM. And the certificate and key will be stored in these storages in an encrypted way. After the certificate and key are stored in these storages, the host cannot read the data from these storages and can only query

the checksum of them. Please note that the certificate or key should not exist in the corresponding storage until it is added to RAM or NVRAM; if it already exists, the host should delete it first, and then add it to the corresponding storage.

AT+QSECWRITE Add a Certificate or Key	
Test Command **AT+QSECWRITE=?**	Response **+QSECWRITE: <file_name>,<file_size>[,(**list of supported **<timeout>)**s**]** **OK**
Read Command **AT+QSECWRITE?**	Response **OK** Or **ERROR**
Write Command **AT+QSECWRITE=<file_name>,<file_size> [,<timeout>]**	Response If format is correct, response: **CONNECT** After the module switches to data mode, the certificate or key data can be input. When the size of the input data reaches **<file_size>** (unit: byte) or the module receives **+++** sequence from UART, the module will return to command mode and reply the following codes: **+QSECWRITE: <upload_size>,<checksum>** **OK** If there is any error: **+CME ERROR: <err>**
Characteristics	The command takes effect immediately. Please also refer to the note below.

Parameter

<file_name>	String format. The name of the file to be stored. The format can be as follows: "RAM:filename" File is uploaded to RAM "NVRAM:filename" File is uploaded to NVRAM. Support server root CA certificate, one client certificate and one client private key. The filename of server root CA certificate must be *CA0*, the filename of client certificate must be *CC0*, and the filename of client private key must be *CK0*. CA0 Identify a server root CA certificate CC0 Identify a client certificate CK0 Identify a client private key
<file_size>	The size of the file to be uploaded. Unit: byte.

	If the file is uploaded to the RAM, the maximum size is 32768. If the file is uploaded to NVRAM, the maximum size is 2017 and the minimum size is 1.
<timeout>	The time in seconds to wait for data input via UART port. Unit: byte. Range: 3-200. The default value is 100.
<upload_size>	The size of the actual uploaded data. Unit: byte.
<checksum>	The checksum of the uploaded data.

NOTE

When the file is uploaded to RAM, the configuration will not be saved. When the file is uploaded to NVRAM, the configuration will be saved automatically.

2.2.3. AT+QSECREAD Query the Checksum of a Certificate or Key

This command is used to query the checksum of a certificate or key. If the checksum is not the same as the original one owned by the user, some mistakes will occur.

AT+QSECREAD Query the Checksum of a Certificate or Key	
Test Command **AT+QSECREAD=?**	Response **+QSECREAD: <file_name>** **OK**
Read Command **AT+QSECREAD?**	Response **OK** Or **ERROR**
Write Command **AT+QSECREAD=<file_name>**	Response **+QSECREAD: <good>,<checksum>** **OK** If some errors occur, response: **+CME ERROR: <err>**
Characteristics	/

Parameter

<file_name>	String format. The name of the file to be stored. The format can be as follows: "RAM:filename" Query the checksum of file that is stored in RAM. "NVRAM:filename" Query the checksum of file that is stored in NVRAM. Support server root CA certificate, one client certificate and one client private key. The filename of server root CA certificate must be

	CA0, the filename of client certificate must be *CC0*, and the filename of client private key must be *CK0*. CA0 Identify a server root CA certificate CC0 Identify a client certificate CK0 Identify a client private key
<good>	Integer type. Indicate whether the certificate or key is correct or not. When uploading the certificate or key by **AT+QSECWRITE**, the checksum of certificate or key will be stored at the same time. After executing **AT+QSECREAD**, the checksum of the certificate or key will be calculated again. Then compare the checksum with the one stored by **AT+QSECWRITE**. If they are the same, the certificate or key is correct; otherwise it is wrong 0 The certificate or key is wrong 1 The certificate or key is correct
<checksum>	The checksum of the file

2.2.4. AT+QSECDEL Delete a Certificate or Key

This command is used to delete a certificate or key.

AT+QSECDEL Delete a Certificate or Key	
Test Command **AT+QSECDEL=?**	Response **+QSECDEL: <file_name>** **OK**
Read Command **AT+QSECDEL?**	Response **OK** Or **ERROR**
Write Command **AT+QSECDEL=<file_name>**	Response **OK** If there is any error: **+CME ERROR: <err>**
Characteristics	/

Parameter

<file_name>	The name of the file to be stored. The format can be as follows: "RAM:filename" Delete a certificate or key that is stored in RAM "NVRAM:filename" Delete a certificate or key that is stored in NVRAM. Support server root CA certificate, one client certificate and one client private key. The filename of server root CA certificate must be *CA0*, the filename of client certificate must be *CC0*, and the

	filename of client private key must be *CK0*.	
	CA0	Identify a server root CA certificate
	CC0	Identify a client certificate
	CK0	Identify a client private key

3 Example

3.1. SSL Function with Certificate and Key in RAM

This is an example about how to set server and client authentication, and the certificate and key are stored in RAM. If the server and client authentication are not needed, please skip this step.

```
//Upload a certificate and key to RAM.
AT+QSECWRITE="RAM:ca_cert.pem",1614,100        //Upload the server root CA certificate to RAM.
CONNECT

<Input the ca_cert.pem data, the size is 1614 bytes>

+QSECWRITE: 1614,4039

OK
AT+QSECWRITE="RAM:client_cert.pem",1419,100     //Upload the client certificate to RAM.
CONNECT

<Input the client_cert.pem data, the size is 1419 bytes>

+QSECWRITE: 1419,618

OK
AT+QSECWRITE="RAM:client_key.pem",1679,100     //Upload the client private key to RAM.
CONNECT

<Input the client_key.pem data, the size is 1679 bytes>

+QSECWRITE: 1679,83a7

OK
```

3.2. SSL Function with Certificate and key in NVRAM

This is an example about how to set server and client authentication, and the certificate and key are stored in NVRAM. If the server and client authentication are not needed, please skip this step.

```
//Upload the certificate and key to NVRAM.
AT+QSECWRITE="NVRAM:CA0",1614,100      //Upload the server root CA certificate to NVRAM.
CONNECT

<Input the CA0 data, the size is 1614 bytes>

+QSECWRITE: 1614,4039

OK
AT+QSECWRITE="NVRAM:CC0",1419,100      //Upload the client certificate to NVRAM.
CONNECT

<Input the CC0 data, the size is 1419 bytes>

+QSECWRITE: 1419,618

OK

AT+QSECWRITE="NVRAM:CK0",1679,100      //Upload the client private key to NVRAM.
CONNECT

<Input the CK0 data, the size is 1679 bytes>

+QSECWRITE: 1679,83a7

OK
```

3.3. Example about SSL Function with HTTPS

3.3.1. Send HTTP GET Response

```
//Step 1: Configure and activate the PDP context.
AT+QIFGCNT=0                           //Set context 0 as foreground context.
OK
AT+QICSGP=1,"CMNET"                    //Set bearer type as GPRS and the APN is "CMNET",
OK                                       which does not have a username and password.
AT+QIREGAPP                            //Register on TCP/IP stack.
```

```
OK
AT+QIACT                                    //Activate GPRS PDP context.
OK
AT+QILOCIP                                  //Query the local IP address.
10.1.83.188

//Step 2: Configure SSL version, cipher suite and there is no authentication.
AT+QSSLCFG="sslversion",1,4                 //Configure SSL version.
OK
AT+QSSLCFG="seclevel",1,2                   //Set the SSL verify level as 1, which means to upload
OK                                          server root CA certificate, client certificate and client private
                                            key by AT+QSECWRITE.
AT+QSSLCFG="ciphersuite",1,"0XFFFF"         //Configure the cipher suite.
OK
AT+QSSLCFG="cacert",1,"RAM:ca_cert.pem"
OK
AT+QSSLCFG="clientcert",1,"RAM:client_cert.pem"
OK
AT+QSSLCFG="clientkey",1,"RAM:client_key.pem"
OK
AT+QSSLCFG="ignorertctime",1                //Ignore the RTC time.
OK

//Step 3: Enable HTTPS function and configure SSL context index for HTTPS.

AT+QSSLCFG="https",1                        //Enable HTTPS function.
OK
AT+QSSLCFG="httpsctxi",1                    //Configure SSL context index as 1.
OK
AT+QHTTPURL=34,60                           //Set the URL.
CONNECT
……

//For example, input 34 bytes: https://124.74.41.170:5008/1K.html.
OK
AT+QHTTPGET=60                              //Send HTTPS GET request.
OK
AT+QHTTPREAD=30                             //Read the response of HTTPS server.
CONNECT
……..                                       //Output the response data of HTTPS server to UART port.
OK
AT+QIDEACT
DEACT OK
```

3.3.2. Send HTTP POST Request

```
//Step 1: Configure and activate the PDP context.
AT+QIFGCNT=0                          //Set context 0 as foreground context.
OK
AT+QICSGP=1,"CMNET"                   //Set bearer type as GPRS and the APN is "CMNET",
OK                                      which does not have a username and password.
AT+QIREGAPP                           //Register on TCP/IP stack.
OK
AT+QIACT                              //Activate GPRS PDP context.
OK
AT+QILOCIP                            //Query the local IP address.
10.1.83.188

//Step 2: Configure SSL version, cipher suite and there is no authentication.
AT+QSSLCFG="sslversion",2,4           //Configure SSL version.
OK
AT+QSSLCFG="seclevel",2,2             //Set the SSL verify level as 2, which means to upload the
OK                                      server root CA certificate, client certificate and client
                                        private key by AT+QSECWRITE.
AT+QSSLCFG="ciphersuite",2,"0XFFFF"   //Configure the cipher suite.
OK
AT+QSSLCFG="cacert",2,"RAM:ca_cert.pem"
OK
AT+QSSLCFG="clientcert",2,"RAM:client_cert.pem"
OK
AT+QSSLCFG="clientkey",2,"RAM:client_key.pem"
OK
AT+QSSLCFG="ignorertctime",1          //Ignore the RTC time.
OK

//Step 3: Enable HTTPS function and configure SSL context index for HTTPS.
AT+QSSLCFG="https",1                  //Enable HTTPS function.
OK
AT+QSSLCFG="httpsctxi",2              //Configure SSL context index as 2.
OK
AT+QHTTPURL=45,60                     //Set the URL.
CONNECT
……
//For example, input 45 bytes: https://220.180.239.212:8011/processorder.php.
OK
AT+QHTTPPOST=48,60,60                 //Send POST data.
CONNECT
……
```

```
//For example, input 48 bytes: Message=1111&Appleqty=2222&Orangeqty=3333&find=1.
OK
AT+QHTTPREAD=30                    //Read the response of HTTPS server.
CONNECT
……..                               //Output the response data of HTTPS server to UART port.
OK
AT+QIDEACT
DEACT OK
```

4 Appendix A References

Table 4: Related Documents

SN.	Document Name	Remark
[1]	GSM 07.07	Digital cellular telecommunications (Phase 2+); AT command set for GSM Mobile Equipment (ME)
[2]	GSM 07.10	GSM 07.10 multiplexing protocol
[3]	Quectel_GSM_HTTP_Application_Note	HTTP application note for GSM modules

Table 5: Terms and Abbreviations

Abbreviation	Description
APN	Access Point Name
CTX	SSL Context
GSM	Global System for Mobile Communications
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
ID	Identification
IP	Internet Protocol
ME	Mobile Equipment
NVRAM	Non Volatile Random Access Memory
PDP	Packet Data Protocol
PPP	Point-to-Point Protocol
RAM	Random Access Memory
RTC	Real-Time Clock

SSL	Security Socket Layer
TCP/IP	Transmission Control Protocol/Internet Protocol
TLS	Transport Layer Security
URL	Uniform Resource Locator