GSM HTTPS

Application Note

GSM/GPRS Module Series

Rev. GSM_HTTPS_Application_Note_V3.2

Date: 2017-01-22

www.quectel.com

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

Quectel Wireless Solutions Co., Ltd.
Office 501, Building 13, No.99, Tianzhou Road, 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/salesupport.aspx

For technical support, or to report documentation errors, please visit:
http://www.quectel.com/support/techsupport.aspx
Or Email to: Support@quectel.com

GENERAL NOTES

QUECTEL OFFERS THE INFORMATION AS A SERVICE TO ITS CUSTOMERS. THE INFORMATION PROVIDED IS BASED UPON CUSTOMERS' REQUIREMENTS. QUECTEL MAKES EVERY EFFORT TO ENSURE THE QUALITY OF THE INFORMATION IT MAKES AVAILABLE. QUECTEL DOES NOT MAKE ANY WARRANTY AS TO THE INFORMATION CONTAINED HEREIN, AND DOES NOT ACCEPT ANY LIABILITY FOR ANY INJURY, LOSS OR DAMAGE OF ANY KIND INCURRED BY USE OF OR RELIANCE UPON THE INFORMATION. ALL INFORMATION SUPPLIED HEREIN IS SUBJECT TO CHANGE WITHOUT PRIOR NOTICE.

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

Contents

Table Index

1 Introduction

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

This document is applicable to Quectel M66, M95, M10, M85 and MC60 modules.

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.

1.1. SSL Version and Cipher Suite

Several SSL versions have been released so far. They are SSL3.0, TLS1.0, TLS1.1 and TLS1.2. The following versions are supported by Quectel modules currently.

Table 1: Supported SSL Versions

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

The following table shows SSL cipher suites supported by Quectel 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 you do not need server and client authentication, 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, you can query the local IP address by AT+QILOCIP command.

Step 4: Configure SSL version, cipher suit, server authentication, client authentication, 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 you failed to activate PDP context by AT+QIACT command, please check the following configurations:

1. Query whether the PS domain is attached or not by AT+CGATT? command. If not, execute AT+CGATT=1 command to attach PS domain.
2. Query the CGREG status by AT+CGREG? command and make sure the PS domain has been registered.
3. Query the PDP context parameters by AT+QIREGAPP command and make sure the APN of specified PDP context has been 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 three PDP contexts activated simultaneously, so you must make sure the amount of activated PDP context is less than 3.

If all above configurations are OK, but you still fail to activate PDP by executing AT+QIACT command, please reboot the module to resolve this issue. After rebooting the module, please check the configurations mentioned above at least three times and each time at an interval of 10 minutes to avoid frequent rebooting the module.

2 Description of AT Command

2.1. AT Command Syntax

Test Command	**AT+<x>=?**	This command returns the list of parameters and value ranges. Set by the corresponding Write Command or internal processes.
Read Command	**AT+<x>?**	This Command returns the currently set value of the parameter or parameters.
Write Command	**AT+<x>=<…>**	This command sets the user-definable parameter values.
Execute Command	**AT+<x>**	This command reads non-variable parameters affected by internal processes in the GSM engine.

2.2. Description of AT Command

2.2.1. AT+QSSLCFG SSL Configuration

This AT command is used to configure the SSL version, cipher suite, secure level, 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. <ctxindex> is the index of the SSL context. Quectel standard modules support six SSL contexts at most. And on the basis of a 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",(0-5),"value"** **OK**
Query settings of the context **AT+QSSLCFG="ctxindex",<ctxindex>**	Response **+QSSLCFG: <ctxindex>,<sslversion>,<seclevel>,<ciphersuite>,<cacert>,<clientcertname>,<clientkeyname>**

	OK Otherwise response **ERROR**
Configure SSL version **AT+QSSLCFG="sslversion",<ctxindex>[,<sslversion>]**	Response **OK** Otherwise response **ERROR** If the third parameter is omitted, query the "sslversion" value. **+QSSLCFG: "sslversion",<sslversion>** **OK**
Configure cipher suite **AT+QSSLCFG="ciphersuite",<ctxindex>[,<**list of supported **<ciphersuite>**s**>]**	Response **OK** Otherwise response **ERROR** If the third parameter is omitted, query the "ciphersuite" value. **+QSSLCFG: "ciphersuite",<ciphersuite>** **OK**
Configure authentication mode **AT+QSSLCFG="seclevel",<ctxindex>[,<seclevel>]**	Response **OK** Otherwise response **ERROR** If the third parameter is omitted, query the "seclevel" value. **+QSSLCFG: "seclevel",<seclevel>** **OK**
Configure the path of root certificate **AT+QSSLCFG="cacert",<ctxindex>[,<cacertname>]**	Response **OK** Otherwise response **ERROR** If the third parameter is omitted, query the "cacertname" value. **+QSSLCFG: "cacert",<cacertname>** **OK**
Configure the path of client certificate **AT+QSSLCFG="clientcert",<ctxindex>[,<clientcertname>]**	Response **OK** Otherwise response

	ERROR If the third parameter is omitted, query the “clientcertname” value. **+QSSLCFG: “clientcert”,<clientcertname>** **OK**
Configure the path of client key **AT+QSSLCFG=“clientkey”,<ctxindex>[,<clientkeyname>]**	Response **OK** Otherwise response **ERROR** If the third parameter is omitted, query the “clientkeyname” value. **+QSSLCFG: “clientkey”,<clientkeyname>** **OK**
Configure whether to ignore the RTC time **AT+QSSLCFG=“ignorertctime”[,<ignorertctime>]**	Response **OK** Otherwise response **ERROR** If the second parameter is omitted, query the “ignorertctime” value. **+QSSLCFG: “ignorertctime”,<ignorertctime>** **OK**
Enable/Disable the HTTPS function **AT+QSSLCFG=“https”[,<httpsenable>]**	Response **OK** Otherwise response **ERROR** If the second parameter is omitted, query the “httpsenable” value. **+QSSLCFG: “https”,<httpsenable>** **OK**
Configure SSL context index for HTTPS **AT+QSSLCFG=“httpsctxi”[,<httpsctxindex>]**	Response **OK** Otherwise response **ERROR** If the second parameter is omitted, query the “httpsctxindex” value.

	+QSSLCFG: "httpsctxi",<httpsctxindex> **OK**
Reference	

Parameter

<ctxindex> SSL context index. Range: 0-5

<sslversion> Configure the supported SSL version
0 SSL3.0
1 TLS1.0
2 TLS1.1
3 TLS1.2
4 All Supported

<ciphersuite> Configuration 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> Configure the authentication mode
0 No authentication
1 Manage server authentication
2 Manage server and client authentication if requested by the remote server

<cacertname> String format, configure the server CA certificate

<clientcertname>String format, configure the client certificate

<clientkeyname> String format, configure the client key

<ignorertctime> Configure whether to ignore the RTC time
0 Do not ignore the RTC time
1 Ignore the RTC time

<httpsenable> Enable/disable the HTTPS function
0 Disable HTTPS
1 Enable HTTPS

<httpsctxindex> Configure the SSL context for HTTPS
<httpsctxindex> is the index of SSL context. If the host does not configure the <httpsctxindex>, the value of <httpsctxindex> is -1. Range: 0-5

NOTES

1. The format of <cacertname>, <clientcertname> and <clientkeyname> can be as follows:
"RAM:filename" File is uploaded to RAM
"NVRAM:filename" File is uploaded to NVRAM. Support two CA certificates, one client certificate

and one client private key. The filename of CA certificate must be CA0 or CA1, the filename of client certificate must be CC0, and the filename of client private key must be CK0.

CA[0,1]	Identify a 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, you need to configure server CA certificate. If both server and client authentications have been set, you need to configure client certificate, server CA certificate and client private key.

2.2.2. AT+QSECWRITE Add a Certificate or Key

This command is used to add user certificate, user key and 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; instead, the host can only query the checksum of them. Please note that before adding a certificate or key to RAM or NVRAM, it should not be existed in the corresponding storage, 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: <filename>,<filesize>[,**list of supported **<timeout>**s**]** **OK**
Read Command **AT+QSECWRITE?**	Response **OK**
Write Command **AT+QSECWRITE=<filename>,<filezsize>[,<timeout>]**	Response If format is correct, response: **Connect** After the module switches to data mode, and the certificate or key data can be input. When the size of the input data reaches <filesize> (unit: byte) or the module receives "+++" sequence from UART, module will return to command mode and reply the following codes: **+QSECWRITE: <uploadsize>,<checksum>** **OK** If some errors occur, response: **+CME ERROR: <err>**
Reference	

Parameter

<filename>	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 two CA certificates, one client certificate and one client private key. The filename of CA certificate must be CA0 or CA1, the filename of client certificate must be CC0, and the filename of client private key must be CK0. CA[0,1] Identify a CA certificate CC0 Identify a client certificate CK0 Identify a client private key
<filesize>	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 2025. 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.
<uploadsize>	The size of the actually uploaded data. Unit: byte
<checksum>	The checksum of the uploaded data

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 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: <filename>** **OK**
Read Command **AT+QSECREAD?**	Response **OK**
Write Command. **AT+QSECREAD=<filename>**	Response **+QSECREAD: <good>,<checksum>** **OK** If some errors occur, response: **+CME ERROR: <err>**
Reference	

Parameter

<filename>	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 two CA certificates, one client certificate and one client private key. The filename of CA certificate must be CA0 or CA1, the filename of client certificate must be CC0, and the filename of client private key must be CK0 CA[0,1] Identify a CA certificate CC0 Identify a client certificate CK0 Identify a client private key
<good>	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, it will calculate the checksum of the certificate or key again, and then compare the checksum with the one stored by AT+QSECWRITE, if they are the same, the certificate or key is correct, otherwise the certificate or key 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: <filename>** **OK**
Read Command **AT+QSECDEL?**	Response **OK**
Write Command **AT+QSECDEL=<filename>**	Response **OK** If some errors occur, response: **+CME ERROR: <err>**
Reference	

Parameter

<filename>	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 two CA certificates, one client certificate and one client private key. The filename of CA certificate must be CA0 or CA1, the filename of client certificate must be CC0, and the filename of client private key must be CK0. CA[0,1] Identify a 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 you do not need server and client authentication, please skip this step.

```
// Upload certificate and key to RAM.
AT+QSECWRITE=“RAM:ca_cert.pem”,1614,100        //Upload the 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 you do not need server and client authentication, please skip this step.

```
//Upload the certificate and key to NVRAM.
AT+QSECWRITE="NVRAM:CA0",1614,100       //Upload the 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", no
OK                                  username and password for the APN.
```

```
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 you should
OK                                              upload CA certificate, client certificate and client private
                                                key by AT+QSECWRITE.
AT+QSSLCFG="ciphersuite",1,"0XFFFF" //Configure 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”, no
OK                                        username and password for the APN.
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 you should
OK                                        upload CA certificate, client certificate and client private
                                          key by AT+QSECWRITE.
AT+QSSLCFG=“ciphersuite”,2,“0XFFFF” //Configure 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 3: 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_AT_Commands_Manual	GSM HTTP AT Commands Manual

Table 4: Terms and Abbreviations

Abbreviation	Description
APN	Access Point Name
HTTPS	Hypertext Transfer Protocol Secure
NVRAM	Non Volatile Random Access Memory
PDP	Packet Data Protocol
PPP	Point-to-Point Protocol
RAM	Random Access Memory
SSL	Security Socket Layer
TCP/IP	Transmission Control Protocol/Internet Protocol
TLS	Transport Layer Security
URL	Uniform Resource Locator