SIM7022 Series_ SSL_Application Note

LPWA Module

SIMCom Wireless Solutions Limited
SIMCom Headquarters Building, Building 3, No. 289 Linhong
Road, Changning District, Shanghai P.R. China
Tel: 86-21-31575100
support@simcom.com
www.simcom.com

Document Title:	SIM7022 Series_SSL_Application Note
Version:	1.00
Date:	2022.12.09
Status:	Released

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

SIMCom Wireless Solutions Limited
SIMCom Headquarters Building, Building 3, No. 289 Linhong 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

About Document

Version History

Revision	Date	Chapter	Description
V1.00	2022.10.24	All	New version

Scope

This document could be applied to following modules.

Name	Type	Size(mm)	Description
SIM7022	NB2	17.6*15.7	Band 1/2/3/4/5/8/12/13/14/17/18/19/20/25/26/28/66/70/85

Contents

1 Introduction

1.1 Purpose of the document

Based on module AT command manual, this document will introduce **SSL** application process for SIM7022 series of module.Developers could understand and develop application quickly and efficiently based on this document.

1.2 SSL versions and cipher suits

The following are SSL versions supported by SIM7022 modules.

Table 1:SSL Versions

SSL Version
SSL3.0
TLS1.1
TLS1.2

The following table shows SSL cipher suites supported by SIM7022 modules.

Table 2:Cipher Suits

Abbreviation	Description
0XC02C	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
0XC030	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
0X009F	TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
0XC0AD	TLS_ECDHE_ECDSA_WITH_AES_256_CCM

0XC09F	TLS_DHE_RSA_WITH_AES_256_CCM
0XC024	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
0XC028	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
0X006B	TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
0XC00A	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
0XC014	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
0X0039	TLS_DHE_RSA_WITH_AES_256_CBC_SHA
0XC0AF	TLS_ECDHE_ECDSA_WITH_AES_256_CCM_8
0XC0A3	TLS_DHE_RSA_WITH_AES_256_CCM_8
0XC02B	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
0XC02F	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
0X009E	TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
0XC0AC	TLS_ECDHE_ECDSA_WITH_AES_128_CCM
0XC09E	TLS_DHE_RSA_WITH_AES_128_CCM
0XC023	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
0XC027	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
0X0067	TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
0XC009	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
0XC013	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
0X0033	TLS_DHE_RSA_WITH_AES_128_CBC_SHA
0XC0AE	TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8
0XC0A2	TLS_DHE_RSA_WITH_AES_128_CCM_8

0X009D	TLS_RSA_WITH_AES_256_GCM_SHA384
0XC09D	TLS_RSA_WITH_AES_256_CCM
0X003D	TLS_RSA_WITH_AES_256_CBC_SHA256
0X0035	TLS_RSA_WITH_AES_256_CBC_SHA
0XC032	TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384
0XC02A	TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384
0XC00F	TLS_ECDH_RSA_WITH_AES_256_CBC_SHA
0XC0A1	TLS_RSA_WITH_AES_256_CCM_8
0X009C	TLS_RSA_WITH_AES_128_GCM_SHA256
0XC09C	TLS_RSA_WITH_AES_128_CCM
0X003C	TLS_RSA_WITH_AES_128_CBC_SHA256
0X002F	TLS_RSA_WITH_AES_128_CBC_SHA
0XC031	TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256
0XC029	TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256
0XC00E	TLS_ECDH_RSA_WITH_AES_128_CBC_SHA
0XC0A0	TLS_RSA_WITH_AES_128_CCM_8
0XC008	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA
0XC012	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA
0X0016	TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA
0X000A	TLS_RSA_WITH_3DES_EDE_CBC_SHA
0XC00D	TLS_ECDH_RSA_WITH_3DES_EDE_CBC_SHA

1.3 The Process of using SSL function

Step 1: Download certificate or private key by AT command **AT+CCERTDOWN**.

Step 2: Config SSL version,authmode,certficate path and so on by AT command **AT+CSSLCFG**.

Step 3: Using **MQTTS**, **HTTPS** or other prtocols calls the corresponding SSL configuration to connect to the remote server.

2 Description of SSL AT Commands

1.4 Detailed Description

1.4.1 AT+CSSLCFG Configure the SSL Context

AT+CSSLCFG Configure the SSL Context	
Test Command AT+CSSLCFG=?	Response **+CSSLCFG: "sslversion",(0-1),(0-4)** **+CSSLCFG: "authmode",(0-1),(0-3)** **+CSSLCFG: "ignorelocaltime",(0-1),(0,1)** **+CSSLCFG: "negotiatetime",(0-1),(10-300)** **+CSSLCFG: "cacert",(0-1),(1-53)** **+CSSLCFG: "clientcert",(0-1),(1-53)** **+CSSLCFG: "clientkey",(0-1),(1-53)** **+CSSLCFG: "enableSNI",(0-1),(0,1)** **OK**
Read Command AT+CSSLCFG?	Response **+CSSLCFG: 0,<sslversion>,<authmode>,<ignoreltime>,<negotiatetime>,<ca_file>,<clientcert_file>,<clientkey_file>,<enableSNI>** **+CSSLCFG: 1,<sslversion>,<authmode>,<ignoreltime>,<negotiatetime>,<ca_file>,<clientcert_file>,<clientkey_file>,<enableSNI>** **OK**
Write Command /*Configure the version of the specified SSL context*/ AT+CSSLCFG="sslversion",<ssl_ctx_index>,<sslversion>	Response 1)If successfully: **OK** 2)If failed: **ERROR**
Write Command /*Configure the authentication mode of the specified SSL context*/ AT+CSSLCFG="authmode",<ssl_ctx_index>,<authmode>	Response 1)If successfully: **OK** 2)If failed: **ERROR**

Write Command /*Configure the ignore local time flag of the specified SSL context*/ **AT+CSSLCFG="ignorelocaltime",<ssl_ctx_index>,<ignoreltime>**	Response 1)If successfully: **OK** 2)If failed: **ERROR**
Write Command /*Configure the negotiate timeout value of the specified SSL context*/ **AT+CSSLCFG="negotiatetime",<ssl_ctx_index>,<negotiatetime>**	Response 1)If successfully: **OK** 2)If failed: **ERROR**
Write Command /*Configure the server root CA of the specified SSL context*/ **AT+CSSLCFG="cacert",<ssl_ctx_index>,<ca_file>**	Response 1)If successfully: **OK** 2)If failed: **ERROR**
Write Command /*Configure the client certificate of the specified SSL context*/ **AT+CSSLCFG="clientcert",<ssl_ctx_index>,<clientcert_file>**	Response 1)If successfully: **OK** 2)If failed: **ERROR**
Write Command /*Configure the client key of the specified SSL context*/ **AT+CSSLCFG="clientkey",<ssl_ctx_index>,<clientkey_file>**	Response 1)If successfully: **OK** 2)If failed: **ERROR**
Write Command /*Configure the enableSNI flag of the specified SSL context */ **AT+CSSLCFG="enableSNI",<ssl_ctx_index>,<enableSNI_flag>**	Response 1)If successfully: **OK** 2)If failed: **ERROR**
Parameter Saving Mode	-
Max Response Time	12000ms
Reference	-

Defined Values

<ssl_ctx_index>	The SSL context ID. The range is 0-1.
<sslversion>	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.
<authmode>	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 reserve
<ignoreltime>	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.
<negotiatetime>	The timeout value used in SSL negotiate stage. The range is 10-300 seconds. The default value is 300.
<ca_file>	The root CA file name of SSL context. The length of filename is from 1 to 53 bytes.
<clientcert_file>	The client cert file name of SSL context. The length of filename is from 1 to 53 bytes.
<clientkey_file>	The client key file name of SSL context. The length of filename is from 1 to 53 bytes.
<enalbeSNI_flag>	The flag to indicate that enable the SNI extension or not, the default value is 1. 0 not enable. 1 enable.

Examples

```
AT+CSSLCFG=?
+CSSLCFG: "sslversion",(0-1),(0-4)
+CSSLCFG: "authmode",(0-1),(0-3)
+CSSLCFG: "ignorelocaltime",(0-1),(0,1)
+CSSLCFG: "negotiatetime",(0-1),(10-300)
+CSSLCFG: "cacert",(0-1),(1-53)
+CSSLCFG: "clientcert",(0-1),(1-53)
+CSSLCFG: "clientkey",(0-1),(1-53)
+CSSLCFG: "enableSNI",(0-1),(0,1)

OK
AT+CSSLCFG?
```

```
+CSSLCFG: 0,4,0,1,300,"","","",1
+CSSLCFG: 1,4,0,1,300,"","","",1

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

1.4.2 AT+CCERTDOWN Download certificate into the module

AT+CCERTDOWN Download certificate into the module	
Test Command AT+CCERTDOWN=?	Response **+CCERTDOWN: (1-53),(1-10240)** **OK**
Write Command AT+CCERTDOWN=<filename>,<len>	Response 1)If parameter is ok and receive enough data within max response time: **><input data>** **OK** 2)If parameter is ok and do not receive enough data within max response time: **><input data>** **ERROR** 2) If parameter is error: **ERROR**
Parameter Saving Mode	-
Max Response Time	120000ms
Reference	-

Defined Values

<filename>	The name of the cacert/clientcert/clientkey file. The length of filename is from 1 to 53 bytes.
<len>	The length of the file data to send. The range is from 1 to 10240 bytes.

Examples

```
AT+CCERTDOWN=?
+CCERTDOWN: (1-53),(1-10240)

OK
AT+CCERTDOWN="baidu.der",889
><input data>

OK
```

1.4.3 AT+CCERTLIST List certificates

AT+CCERTLIST List certificates	
Test Command **AT+CCERTLIST=?**	Response **OK**
Execute Command **AT+CCERTLIST**	Response **+CCERTLIST: <filename>** **…** **+CCERTLIST: <filename>** **OK**
Parameter Saving Mode	-
Max Response Time	12000ms
Reference	-

Defined Values

<filename>	The name of the cacert/clientcert/clientkey file. The length of filename is from 1 to 53 bytes.

Examples

```
AT+CCERTLIST
+CCERTLIST: "baidu.der"

OK
```

1.4.4 AT+CCERTDELE Delete certificates

AT+CCERTDELE Delete certificates	
Test Command AT+CCERTDELE=?	Response **OK**
Write Command AT+CCERTDELE=<filename>	Response 1) If remove the file successfully: **OK** 2) Else **ERROR**
Parameter Saving Mode	-
Max Response Time	12000ms
Reference	-

Defined Values

<filename>	The name of the cacert/clientcert/clientkey file. The length of filename is from 1 to 53 bytes.

Examples

```
AT+CCERTLIST
+CCERTLIST: "baidu.der"

OK
AT+CCERTDELE="baidu.der "
OK
AT+CCERTLIST
OK
```

3 Examples

1.5 Set to verify the server for first SSL context

```
AT+CCERTDOWN=”baidu.der”,889                    // download the cacert
> 0?u0?]?
    KZ脭0
    *咹噍
0W1
0   UBE10U
GlobalSign nv-sa10U
Root CA10UGlobalSign Root CA0-
980901120000Z
280128120000Z0W1
0   UBE10U
GlobalSign nv-sa10U
Root CA10UGlobalSign Root CA0?"0
    *咹噍
 ?0?
? ?鏅嶋 c O姙   媰%k闔?肮?金c砚gf?菹H+顛墡
毌)€e玐?双Lp?
0? 峅    詫P颬钅 .軌 蹃 驊}郷 ?0 巀 Cs 骍 檎 j
悴?V98o<�[*M脓T竘壥浚<叔罷?<扶挈躷n摛a禅
犇瞰嗚娬ml誼暍槿
8腧$OsT???4祴尯w嫠?  懌    Sn   愇   {7t   箕
G?Qcy缤瓬&?+袴銛謉*變4   ,*殶CJ呯觸?h铘
騌?犇?    0@0U   0U   0   0U`{fE
検埘/}??   K0
    *咹噍
 ?  諔鍚Ov祖快孩??2�?+?瀆縦^?H跺 3 ?aM
覭   ?腞錼U蚝锃9酙? f /?;餇PV
?湍
pQ敍?呙_所?A滙]ud
  U0鞀?
?篃F霆   糹 ?Hd氝轡︺)???   i,i$x-厂qb钍葥?]
婷鳪唍*腣1暠g墔+鶓   F?獋錽Q輕粉V=a鍃醆嘱?
轆?豯R緭S?槅   鈪_呉A踞??   o  赊?謋U恻
H?&i?
```

```
OK
AT+CCERTLIST                              // show the cacert in the module
+CCERTLIST: “baidu.der”

OK
AT+CSSLCFG=”cacert”,0,”baidu.der”         // set the cacert for first SSL context
OK
AT+CSSLCFG=”authmode”,0,1                 // set to verify the server for first SSL context
OK
```

1.6 Set to verify the server and client for first SSL context

```
AT+CCERTDOWN=”cacert.crt”,1208            // download the cacert
>-----BEGIN CERTIFICATE-----
MIIDUDCCAjgCCQD695Erl0p93DANBgkqhkiG
9w0BAQsFADBpMQswCQYDVQQGEwJD
TjESMBAGA1UECAwJR3Vhbmdkb25nMRIwE
AYDVQQHDAlHdWFuZ3pob3UxDTALBgNV
BAoMBFlaWk4xCzAJBgNVBAsMAlJKMRYwF
AYDVQQDDA1tcXR0Mi55enpuLmNuMCAX
DTIxMDMxMTAxNTY0MVoYDzIxMjEwMjE1MD
E1NjQxWjBpMQswCQYDVQQGEwJDTjES
MBAGA1UECAwJR3Vhbmdkb25nMRIwEAYD
VQQHDAlHdWFuZ3pob3UxDTALBgNVBAoM
BFlaWk4xCzAJBgNVBAsMAlJKMRYwFAYDV
QQDDA1tcXR0Mi55enpuLmNuMIIBIjAN
BgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCA
QEA2MQrt3DUsLIqQyDWVveNWRWrleTb
f43qc39hG5Xb0AVSm5vBzfrNb6uL9IMIQNns
OrAhXa62mbrg4MU0QdALAGKVf9+f
ORLxUnbGxDHVPYUWWHDBliwxENe/w53GG
ylDc+y5cmq1toiZw0bvUyj5ziZyAEJD
9l2Cd2DbPCYXmGwj9FqPWPpzPlGan9EgHx
JoUVtAFskfv8Zo2ilGcC4K3oq7AelU
rc1n0wH9kaa86WRf8m+HjTZSPAuM/gC9+Cq
NF83BCMbGayishvQRQNo0+4l7Zywp
qO5kHM1FJQiMWcmjyDfDhXL4cbSs+y7h3s8
aHkolSwE+39dlaNcET9GzawIDAQAB
MA0GCSqGSIb3DQEBCwUAA4IBAQARIHuyu
ZmRkfcy2DFG9sVQ6+YvoQ1YtHCKsmH6
```

```
Wa/NSNay5P+AciZorTyM5P0TbI2honrvrE+fSj
+VHMCQvxmtCMiE5w3Zx/cCM3ZU
vPR2qj4AotvNKLbwoxYIu4+YCJVngWJSLsq
NWV9+yZH8KuNtQ8MVQXTkI2Q94Yux
sqyfzEi3ZatpAH0QoERCkQBajJYzeJljstLuah
Be0PgGBkOmwUHydEdjOfEesBG3
0/D0g9kjYIv407nziXqiOpGVG+S6NaPD7DcaV
XQOrohPg8eA2bDI3a207pIWqgYS
6toKCa+YUM0mKLVGJYYYOe2PN5GB8SiVG
MhtecacvyLWqOGL
-----END CERTIFICATE-----

OK
AT+CCERTDOWN=”clientcert.crt”,1324          // download the clientcert
>-----BEGIN CERTIFICATE-----
MIIDITCCAn0CFB+2lerBWIOgGvj5h7cZ9ZBe/
6s1MA0GCSqGSIb3DQEBCwUAMIGQ
MQswCQYDVQQGEwJDTjESMBAGA1UECAw
JQ2hvbmdxaW5nMQ4wDAYDVQQHDAVDaG9
u
ZzEPMA0GA1UECgwGU0INQ09NMREwDwYD
VQQLDAhTb2Z0d2FyZTEYMBYGA1UEAwwP
eW9uZ2hhbmcuZG9tYWluMR8wHQYJKoZIhv
cNAQkBFhB5b25naGFuZ0AxNjMuY29t
MB4XDTIwMTIxODA2MjUwNloXDTIxMDExNz
A2MjUwNlowfTELMAkGA1UEBhMCQ04x
DjAMBgNVBAgMBWNoYW9uMQ4wDAYDVQ
QHDAVhc2R3YTENMAsGA1UECgwEZHdhZD
EO
MAwGA1UECwwFd2FzZHcxGDAWBgNVBAM
MD3lvbmdoYW5nLmRvbWFpbjEVMBMGCSq
G
SIb3DQEJARYGZHdhc2R3MIIBIjANBgkqhkiG
9w0BAQEFAAOCAQ8AMIIBCgKCAQEA
03lyavvX/IlbJ43+7cSeosaPnvsZ4I/0LZSVhrpx
uyNAvrki1DdjL58Z8syRqDCq
L/5GcIxp0hvfphtBzAFgsYBX/CMUGxRIkF0Uu
xRXFiaXS96zudvVqInzpo6pBMcs
FelidYRGP7WygKyJ3s41NzFpPvMBIZGJM9ZI
e4jHb8kc8o5oHBIo+IC+PucKqLH1
/t8dpr2c9DToRMYQAtIbpzfqkbzy0vpHzBy5y6
FNMJ5XbsVVhfAXg+hM5VEdKguJ
V3PDVCLDNdd6kL/CY04weZP5qB1gGRpt0Tq
faiyMSbSJZHx2TX5MAFc9PwogFoAX
LCtNrHccJAgSUJjchwl/oQIDAQABMA0GCSq
```

```
GSIb3DQEBCwUAA4IBAQAI/JWAUVHP
6HpUycrxAykjtHq+nBq2VhsrMPf3h7PuDsKK
qpk6QWs1Q5Nni0EzHL4m7k3osaTM
e45mEQOmyCjZaHkBNv2g9iaKjXB+Vj5pV5AT
sZWJxrsQ2Y4/7vnDWkd42s5a02ee
k9pDSOg9uW0L/2mbHyvcEvuUmi13xkkKZ95
Lb1pSR/iDRIPMLLgfCyBvKZfypoK6
SL66SAtrgDUJN1yMflidzT6kmDfZ9+zvqcDQB
QPp/IK28AkHKwk0WX9IlbUAxHKW
acUIhgPI7/80ZfDWt/gqa5ALXAJXaevjD4VPv0
OGux/H7c9H335+xsI+OnpmPeEh
kTiks3sCOgCK
-----END CERTIFICATE-----

OK
AT+CCERTDOWN=”clientkey.key”,1706          // download the clientkey.key
>-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEA03lyavvX/IlbJ43+7cSeos
aPnvsZ4I/0LZSVhrpxuyNAvrki
1DdjL58Z8syRqDCqL/5GcIxp0hvfphtBzAFgs
YBX/CMUGxRlkF0UuxRXFiaXS96z
udvVqlnzpo6pBMcsFelidYRGP7WygKyJ3s41
NzFpPvMBIZGJM9ZIe4jHb8kc8o5o
HBlo+IC+PucKqLH1/t8dpr2c9DToRMYQAtIbp
zfqkbzy0vpHzBy5y6FNMJ5XbsVV
hfAXg+hM5VEdKguJV3PDVCLDNdd6kL/CY04
weZP5qB1gGRpt0TqfaiyMSbSJZHx2
TX5MAFc9PwogFoAXLCtNrHccJAgSUJjchwl/
oQIDAQABAoIBAF+zRB46HkMSePfr
gIRISEztgq01gT86oSvHY+t2kGVZzMEC00oqT
o6Md5ezD++HJl1VutbQFEtrJcjr
6NjDftiU9jw6O60Ni/DKzsjiqY5ypGFHRRpE6+
qNjn+/a6mn4MF+do2r8laAWWl1
q2bS9q4lhDVij0L6e5aj0GVnPqnNDI823QMh/Y
W96yUMoR2Ba85eqSZ+szCEJ9aC
13XC0gtYsjcw8KmjlnwEU7bqlYh8oSZxj4NcO
YbotWjFwNuXgW8C175N0aEYqq54
W95lJU91mvGflmFQezZlPoy/rFMskxv4uyza0
3Ri0ZsoLl2FzdRKCXMVRv1pH4h7
tsj9rOECgYEA9bGs4WYF45wvc+lVj1yl+w+wk
BYKjlayVfdewY6BFab5TSmKg/O/
tjl+Asg2Zg0t9zfegu5hL8MKOTmlQJplkavESs
wzaDILkX6vURVizusU5ZwTs9f5
3NCfP9xLiKW3xGzAfwZ4WE7nfdQpnZQazT/
OkloiJI5EU+EW/CdKwHUCgYEA3FhP
```

```
/kK8j1HTgEY0q/q01mgXVnevnTxDVRSQ+xET
01F65xam0x2wlK5oELz3ebecJFF2
yHe6BZMrRr2L100p3mjkA6onq2clLodpjb+BH
BVMhwLaWori9z8P7ZSyNCTD3dbz
yT1E9f724uB6QywYF5ULdSLMnp7TtFb/7GIY
XP0CgYEA0yTTHo1v6DA0M5fF2MM8
UQ4IvV3DyQqEvg4tV4fg9TueqapWiJI0Gt7IndI
zrnYLF5bi2YCE8ufZpF4e3wsQ
2IRV17XvQ88mU+4cOkF3vb0XI0/jOr4T06ISAi
6OlytbZynSsBdeWv3MQT2QWgSK
I/MK0ok1KFc+7xrUhvQ5cHUCgYBi2behUJ05
CrOAs35DvShNm8sEfpMpTfTDAYP9
Lm8feUISzKWwxnwGZ6vF/pBjaYzB+k34p0Wr
5JcgmD5ZK4PiBFpujnJXgeF7W0Ju
VgB88c0wMIZ24iHqW78wjWnY3LmGFz2tBTC
fz29A3wXahriUM8g9F4yGiKhfGjyb
9nIJnQKBgQDLp1R4OPqHIsO9CTgQSwmI88
6xb+dxDn+e1zjm26X8zg5UgSbZiP+8
0Xc0xwxtyBiAe9nd/QPjXoq4g2kQM8tvHuOITt
iWyPyehmh+7owquR5lLoiymK37
zb1Oc1xIC8aVKRCwppMTtWEaE1dtlU4ZprIq
YdLjCffFXg4G2H9/Yg==
-----END RSA PRIVATE KEY-----

OK
AT+CCERTLIST                                   // show the cacert in the module
+CCERTLIST: “cacert.crt”
+CCERTLIST: “clientcert.crt”
+CCERTLIST: “clientkey.key”

OK
AT+CSSLCFG=”cacert”,0,”cacert.crt”             // set the cacert for first SSL context
OK
AT+CSSLCFG=”clientcert”,0,”clientcert.crt”     // set the clientcert for first SSL context
OK
AT+CSSLCFG=”clientkey”,0,”clientkey.key”       // set the clientkey for first SSL context
OK
                                               // set to verify the server and client for first SSL
AT+CSSLCFG=”authmode”,0,2
                                               context
OK
```

4 Appendix A References

Table 3:Related Documents

SN	Document Name	Remark
[1]	SIM7022 Series_AT Command Manual	AT Command of SIM7022 module

Table 4:Terms and Abbreviations

Abbreviation	Description
ME	Mobile Equipment
MS	Mobile Station
TA	Terminal Adapter
DCE	Data Communication Equipment
TE	Terminal Equipment
DTE	Data Terminal Equipment
PDP	Packet Data Protocol
TCP	Terminal Control Protocol
UDP	User Datagram Protocol