

[toc]

# 课下测试

## ch03

### 作业题目：TLCP协议验证

完成下面任务（99分）

1. 参考云班课课程资源中“ch03 密码技术-协议验证”的“VPN协议验证.pdf”和“ch03 密码技术-09-密码协议验证TLCP.mp4”，对 TLCP 协议进行验证，提交验证过程。（79分）
2. 不使用 “VPN协议验证.pdf”中的商用密码工具，使用 OpenSSL或者 GmSSL命令进行验证，提交验证过程（选做,加分项目，加20分）

### 作业提交要求 (1')

0. 记录实践过程和 AI 问答过程，尽量不要截图，给出文本内容
  1. (选做)推荐所有作业托管到 [gitee](#)或 [github](#) 上
  2. (必做)提交作业 markdown文档，命名为“学号-姓名-作业题目.md”
  3. (必做)提交作业 markdown文档转成的 PDF 文件，命名为“学号-姓名-作业题目.pdf”
- [github链接](#)

### 作业内容

#### 签名验证

##### 服务端密钥交换消息验证

- client random("client hollo")

6541f880b28ee1eaea9d5a92679ce16cea85c56df7ede810ffa78d44fb0b944d.

- server random("server hollo")

553c83b288b79abae1841fe1e05ff29d0771d3160314d562444f574e47524400

- server加密证书(server发来的"certificate"中的第二个证书)

0001f7308201f330820198a00302010202145326ccffbb03ba78b831599da3e085133de83da8300a06082a811ccf550183753068310b300906035504061302434e3111300f06035504080c085368616e646f6e67310f300d060355040a0c0653616e7365633122302006092a864886f70d01090116136e656967686261647340676d61696c2e636f6d3111300f06035504030c08736d325f

```
726f6f74301e170d3230303730323036303030355a170d3330303633303036303030355a306e
310b300906035504061302434e3111300f06035504080c085368616e646f6e67310f300d0603
55040a0c0653616e7365633122302006092a864886f70d01090116136e656967686261647340
676d61696c2e636f6d3117301506035504030c0e736d325f7365727665725f656e6330593013
06072a8648ce3d020106082a811ccf5501822d0342000499fe11af63e5cc352759771030cf5a
b549f4f2ee3ccd8126345d81d0116aa15087977e9303701cc565ecae91441b670af28eade4c
579ef817f2ad65d054fda2a31a301830090603551d1304023000300b0603551d0f0404030204
70300a06082a811ccf550183750349003046022100fb2cfe11bdacf92a9f2b10bfdb9e6d7fff
08e8683ee63b440b25c5c6836f85c5022100b66a71e1236de75383976a5e8d38f06f0db145ae
01ea8313231edfe35f215639
```

- 签名原文 = 32位的客户端随机数 + 32位的服务器随机数 + 服务端加密证书

```
6541f880b28ee1eaea9d5a92679ce16cea85c56df7ede810ffa78d44fb0b944d
553c83b288b79abae1841fe1e05ff29d0771d3160314d562444f574e4752440
00001f7
308201f330820198a00302010202145326ccffbb03ba78b831599da3e085133de83da8300a06
082a811ccf550183753068310b300906035504061302434e3111300f06035504080c08536861
6e646f6e67310f300d060355040a0c0653616e7365633122302006092a864886f70d01090116
136e656967686261647340676d61696c2e636f6d3111300f06035504030c08736d325f726f6f
74301e170d3230303730323036303030355a170d3330303633303036303030355a306e310b30
0906035504061302434e3111300f06035504080c085368616e646f6e67310f300d060355040a
0c0653616e7365633122302006092a864886f70d01090116136e656967686261647340676d61
696c2e636f6d3117301506035504030c0e736d325f7365727665725f656e633059301306072a
8648ce3d020106082a811ccf5501822d0342000499fe11af63e5cc352759771030cf5ab549f4
f2ee3ccd8126345d81d0116aa15087977e9303701cc565ecae91441b670af28eade4c579ef8
17f2ad65d054fda2a31a301830090603551d1304023000300b0603551d0f040403020470300a
06082a811ccf550183750349003046022100fb2cfe11bdacf92a9f2b10bfdb9e6d7fff08e868
3ee63b440b25c5c6836f85c5022100b66a71e1236de75383976a5e8d38f06f0db145ae01ea83
13231edfe35f215639
```

- 原文的签名值的DER编码(在server发来的"key exchange"中)

```
3046022100
d111933a5d9b42e7010e8787dcc6c28eda4e853fde9b0a27f8aa5937d091afdc
022100
8dd88b5116738935cefad8b5cd7a5f7b554ba92cd20d22e722a055f4fc2e4d61
```

- 解码后得到真实的签名值

```
d111933a5d9b42e7010e8787dcc6c28eda4e853fde9b0a27f8aa5937d091afdc8dd88b511673
8935cefad8b5cd7a5f7b554ba92cd20d22e722a055f4fc2e4d61
```

- 公钥(在签名证书即"certificate"中的第一个证书中，格式04||x||y)

04  
9cd6f179df6a5fd2b71f2a772764b4fc54681c8444285421bdcdf81eee4baa3d95e50b04a592  
21118da1e0aac92572bea750db3d7eae9765ebfda7110fc4e7d7

- 验证签名
  - 验证网站
  - 实际使用时公钥前的'04'要去掉

🔍 签名验签 ×

签名验签数据

6541f880b28ee1eaea9d5a92679ce16cea85c56df7ede810ffa78d44fb0b944d553c83b288b79abae1841fe1e05ff29d0771d3160314d562444f574e47524400001f7308201f330820198a00302010202145326ccffbb03ba78b831599da3e085133de83da8300a06082a811ccf550183753068310b300906035504061302434e3111300f06035504080c085368616e646f6e67310f300d060355040a0c0653616e7365633122302006092a864886f70d01090116136e656967686261647340676d61696c2e636f6d3111300f06035504030c08736d325f726f6f74301e170d3230303730323036303030355a170d3330303633303036303030355a306e31

密钥

9cd6f179df6a5fd2b71f2a772764b4fc54681c8444285421bdcdf81eee4baa3d95e50b04a59221118da1e0aac92572bea750db3d7eae9765ebfda7110fc4e7d7

签名值

d111933a5d9b42e7010e8787dcc6c28eda4e853fde9b0a27f8aa5937d091afdc8dd88b5116738935cefad8b5cd7a5f7b554ba92cd20d22e722a055f4fc2e4d6l

签名

验签

结果

验证成功

客户端证书和密钥交换验证

- 整个"client hello"值

0100003c01016541f880b28ee1eaea9d5a92679ce16cea85c56df7ede810ffa78d44fb0b944d00000ae013e011e003e00100ff0100000900230000000f000101

- 整个"server hello"值

0200004d0101553c83b288b79abae1841fe1e05ff29d0771d3160314d562444f574e4752440020f9fd8a5e100eb5f3fb89e091370a8ec0f2a660348bfa82387d9c25c21b6c24f3e013000005ff01000100

- server 双证书("Certificate"中的"Handshake Protocol: Certificate"值)

```
0b0003f20003ef0001f2
30
8201ee
30820194a00302010202145326ccffbb03ba78b831599da3e085133de83da7300a06082a811c
cf550183753068310b300906035504061302434e3111300f06035504080c085368616e646f6e
67310f300d060355040a0c0653616e7365633122302006092a864886f70d01090116136e6569
67686261647340676d61696c2e636f6d3111300f06035504030c08736d325f726f6f74301e17
0d3230303730323036303030355a170d3330303633303036303030355a306a310b3009060355
04061302434e3111300f06035504080c085368616e646f6e67310f300d060355040a0c065361
6e7365633122302006092a864886f70d01090116136e656967686261647340676d61696c2e63
6f6d3111300f06035504030c0a736d325f7365727665723059301306072a8648ce3d02010608
2a811ccf5501822d034200049cd6f179df6a5fd2b71f2a772764b4fc54681c8444285421bdcd
f81eee4baa3d95e50b04a59221118da1e0aac92572bea750db3d7eae9765ebfda7110fc4e7d7
a31a301830090603551d1304023000300b0603551d0f0404030205e0300a06082a811ccf5501
83750348003045022100dd112f544c8699205fb12e35a07065fb916d6e95445f8870de27ad14
67aeabf402201b02357965886bdbcf2407ad1c42bc8d4fe9d686c064aea52dc31f1ca931736b
0001f7308201f330820198a00302010202145326ccffbb03ba78b831599da3e085133de83da8
300a06082a811ccf550183753068310b300906035504061302434e3111300f06035504080c08
5368616e646f6e67310f300d060355040a0c0653616e7365633122302006092a864886f70d01
090116136e656967686261647340676d61696c2e636f6d3111300f06035504030c08736d325f
726f6f74301e170d3230303730323036303030355a170d3330303633303036303030355a306e
310b300906035504061302434e3111300f06035504080c085368616e646f6e67310f300d0603
55040a0c0653616e7365633122302006092a864886f70d01090116136e656967686261647340
676d61696c2e636f6d3117301506035504030c0e736d325f7365727665725f656e6330593013
06072a8648ce3d020106082a811ccf5501822d0342000499fe11af63e5cc352759771030cf5a
b549f4f2ee3ccd8126345d81d0116aa15087977e9303701cc565ecae91441b670af288eade4c
579ef817f2ad65d054fda2a31a301830090603551d1304023000300b0603551d0f0404030204
70300a06082a811ccf550183750349003046022100fb2cfe11bdacf92a9f2b10bfdb9e6d7fff
08e8683ee63b440b25c5c6836f85c5022100b66a71e1236de75383976a5e8d38f06f0db145ae
01ea8313231edfe35f215639
```

- server key exchange("Server Key Exchange"中的"Handshake Protocol: Server Key Exchange")

```
0c00004a00483046022100d111933a5d9b42e7010e8787dcc6c28eda4e853fde9b0a27f8aa59
37d091afdc0221008dd88b5116738935cefad8b5cd7a5f7b554ba92cd20d22e722a055f4fc2e
4d61
```

- certificate request(证书请求通常包含公钥和相关信息（如组织名称、域名、国家等），CA会使用这些信息来生成并签名证书。)

```
0d0000050201020000
```

- server hello done

```
0e000000
```

- client 双证书

```
0b0003f10003ee0001f2
30
8201ee
30820194a00302010202145326ccffbb03ba78b831599da3e085133de83da5300a06082a811c
cf550183753068310b300906035504061302434e3111300f06035504080c085368616e646f6e
67310f300d060355040a0c0653616e7365633122302006092a864886f70d01090116136e6569
67686261647340676d61696c2e636f6d3111300f06035504030c08736d325f726f6f74301e17
0d3230303730323036303030355a170d3330303633303036303030355a306a310b3009060355
04061302434e3111300f06035504080c085368616e646f6e67310f300d060355040a0c065361
6e7365633122302006092a864886f70d01090116136e656967686261647340676d61696c2e63
6f6d3111300f06035504030c0a736d325f636c69656e743059301306072a8648ce3d02010608
2a811ccf5501822d03420004f1351cf6292f6f4ca4a4b727d81a3285826ce231c169bbb301b5
52991de5d016fe552ec26aef8260ca85f2007b9ab836be6a61edf3d9c8567dcce1d1394dcfa9
a31a301830090603551d1304023000300b0603551d0f0404030205e0300a06082a811ccf5501
83750348003045022100b8b4f96c8644b8c8ebae6cd333ecc5df783ed7419bf46ad92e51e725
7d25670b0220294db4ed04feebeec47f95048c4d2c13781f1ac930e3756a6d8076ffc1352ec8
0001f6308201f230820198a00302010202145326ccffbb03ba78b831599da3e085133de83da6
300a06082a811ccf550183753068310b300906035504061302434e3111300f06035504080c08
5368616e646f6e67310f300d060355040a0c0653616e7365633122302006092a864886f70d01
090116136e656967686261647340676d61696c2e636f6d3111300f06035504030c08736d325f
726f6f74301e170d3230303730323036303030355a170d3330303633303036303030355a306e
310b300906035504061302434e3111300f06035504080c085368616e646f6e67310f300d0603
55040a0c0653616e7365633122302006092a864886f70d01090116136e656967686261647340
676d61696c2e636f6d3117301506035504030c0e736d325f636c69656e745f656e6330593013
06072a8648ce3d020106082a811ccf5501822d0342000449bb9eefc8831883173196e6c9de37
fefc24251429f3bd5950a11f60fbf016b573bbece3ce92c624b21cb570fb55fe5b1c1e74cceb
e871ff7d4503128da4ef7ea31a301830090603551d1304023000300b0603551d0f0404030204
70300a06082a811ccf550183750348003045022100ddef0028f0e53dd09b7d32b93426045def
0b7524725ce025958c96e9a617a167022015e30a9269fa4eecef991a1935837abfe47a6970f9
5829ad9cc518133997d83c
```

- client key exchange

```
1000009e009c30819902210082726f634e53e6c54884ef29ae668e6dd54ee9e32e7aafe5b2fb
15ddf5aadd4502206f3b2e27716ea92f4d68b48de9645dc949bfa5da757a0b15f42262ef17c0
85b30420af9da5257ed27b586587425dc755b7886bdf55d0866da22f56c17d8e7f1468740430
07194ce4335592c2a674ce3f15914a2ea04f5c64ddf25b796a4cb7df1278b413e31c02662554
63b741dddff834188195a
```

- 组合数据(规则: clienthello||serverhello||server证书||serverkeyexchange||servercertificaterequest||serverhellodone||client证书||clientkeyexchange)

```
0100003c01016541f880b28ee1eaea9d5a92679ce16cea85c56df7ede810ffa78d44fb0b944d
00000ae013e011e003e00100ff0100000900230000000f0001010200004d0101553c83b288b7
9abae1841fe1e05fff29d0771d3160314d562444f574e4752440020f9fd8a5e100eb5f3fb89e0
91370a8ec0f2a660348bfa82387d9c25c21b6c24f3e013000005fff010001000b0003f20003ef
```

0001f2308201ee30820194a00302010202145326ccffbb03ba78b831599da3e085133de83da7  
300a06082a811ccf550183753068310b300906035504061302434e3111300f06035504080c08  
5368616e646f6e67310f300d060355040a0c0653616e7365633122302006092a864886f70d01  
090116136e656967686261647340676d61696c2e636f6d3111300f06035504030c08736d325f  
726f6f74301e170d3230303730323036303030355a170d3330303633303036303030355a306a  
310b300906035504061302434e3111300f06035504080c085368616e646f6e67310f300d0603  
55040a0c0653616e7365633122302006092a864886f70d01090116136e656967686261647340  
676d61696c2e636f6d3113301106035504030c0a736d325f7365727665723059301306072a86  
48ce3d020106082a811ccf5501822d034200049cd6f179df6a5fd2b71f2a772764b4fc54681c  
8444285421bdcdf81eee4baa3d95e50b04a59221118da1e0aac92572bea750db3d7eae9765eb  
fda7110fc4e7d7a31a301830090603551d1304023000300b0603551d0f0404030205e0300a06  
082a811ccf550183750348003045022100dd112f544c8699205fb12e35a07065fb916d6e9544  
5f8870de27ad1467aeabf402201b02357965886bdbcf2407ad1c42bc8d4fe9d686c064aea52d  
c31f1ca931736b0001f7308201f330820198a00302010202145326ccffbb03ba78b831599da3  
e085133de83da8300a06082a811ccf550183753068310b300906035504061302434e3111300f  
06035504080c085368616e646f6e67310f300d060355040a0c0653616e736563312230200609  
2a864886f70d01090116136e656967686261647340676d61696c2e636f6d3111300f06035504  
030c08736d325f726f6f74301e170d3230303730323036303030355a170d3330303633303036  
303030355a306e310b300906035504061302434e3111300f06035504080c085368616e646f6e  
67310f300d060355040a0c0653616e7365633122302006092a864886f70d01090116136e6569  
67686261647340676d61696c2e636f6d3117301506035504030c0e736d325f7365727665725f  
656e633059301306072a8648ce3d020106082a811ccf5501822d0342000499fe11af63e5cc35  
2759771030cf5ab549f4f2ee3ccd8126345d81d0116aa15087977e9303701cc565ecae91441b  
670af288eade4c579ef817f2ad65d054fda2a31a301830090603551d1304023000300b060355  
1d0f040403020470300a06082a811ccf550183750349003046022100fb2cfe11bdacf92a9f2b  
10bfdb9e6d7fff08e8683ee63b440b25c5c6836f85c5022100b66a71e1236de75383976a5e8d  
38f06f0db145ae01ea8313231edfe35f2156390c00004a00483046022100d111933a5d9b42e7  
010e8787dcc6c28eda4e853fde9b0a27f8aa5937d091afdc0221008dd88b5116738935cefad8  
b5cd7a5f7b554ba92cd20d22e722a055f4fc2e4d610d00000502010200000e000000b0003f1  
0003ee0001f2308201ee30820194a00302010202145326ccffbb03ba78b831599da3e085133d  
e83da5300a06082a811ccf550183753068310b300906035504061302434e3111300f06035504  
080c085368616e646f6e67310f300d060355040a0c0653616e7365633122302006092a864886  
f70d01090116136e656967686261647340676d61696c2e636f6d3111300f06035504030c0873  
6d325f726f6f74301e170d3230303730323036303030355a170d333030363330303630303035  
5a306a310b300906035504061302434e3111300f06035504080c085368616e646f6e67310f30  
0d060355040a0c0653616e7365633122302006092a864886f70d01090116136e656967686261  
647340676d61696c2e636f6d3113301106035504030c0a736d325f636c69656e743059301306  
072a8648ce3d020106082a811ccf5501822d03420004f1351cf6292f6f4ca4a4b727d81a3285  
826ce231c169bbb301b552991de5d016fe552ec26aef8260ca85f2007b9ab836be6a61edf3d9  
c8567dcce1d1394dcfa9a31a301830090603551d1304023000300b0603551d0f0404030205e0  
300a06082a811ccf550183750348003045022100b8b4f96c8644b8c8ebae6cd333ecc5df783e  
d7419bf46ad92e51e7257d25670b0220294db4ed04feebeec47f95048c4d2c13781f1ac930e3  
756a6d8076ffc1352ec80001f6308201f230820198a00302010202145326ccffbb03ba78b831  
599da3e085133de83da6300a06082a811ccf550183753068310b300906035504061302434e31  
11300f06035504080c085368616e646f6e67310f300d060355040a0c0653616e736563312230  
2006092a864886f70d01090116136e656967686261647340676d61696c2e636f6d3111300f06  
035504030c08736d325f726f6f74301e170d3230303730323036303030355a170d3330303633  
303036303030355a306e310b300906035504061302434e3111300f06035504080c085368616e  
646f6e67310f300d060355040a0c0653616e7365633122302006092a864886f70d0109011613  
6e656967686261647340676d61696c2e636f6d3117301506035504030c0e736d325f636c6965  
6e745f656e633059301306072a8648ce3d020106082a811ccf5501822d0342000449bb9eefc8  
831883173196e6c9de37fefc24251429f3bd5950a11f60fbf016b573bbece3ce92c624b21cb5  
70fb55fe5b1c1e74ccebe871ff7d4503128da4ef7ea31a301830090603551d1304023000300b  
0603551d0f040403020470300a06082a811ccf550183750348003045022100ddef0028f0e53d

```
d09b7d32b93426045def0b7524725ce025958c96e9a617a167022015e30a9269fa4eecef991a
1935837abfe47a6970f95829ad9cc518133997d83c1000009e009c30819902210082726f634e
53e6c54884ef29ae668e6dd54ee9e32e7aafe5b2fb15ddf5aadd4502206f3b2e27716ea92f4d
68b48de9645dc949bfa5da757a0b15f42262ef17c085b30420af9da5257ed27b586587425dc7
55b7886bdf55d0866da22f56c17d8e7f146874043007194ce4335592c2a674ce3f15914a2ea0
4f5c64ddf25b796a4cb7df1278b413e31c0266255463b741dddf834188195a
```

- 签名数据(由上面的组合的数据进行sm3 hash计算得到)

```
c08d85599e2172797add1a3d510b63f3122815b9716e426facd0ca83a36c60cf
```

- 获取客户端的签名值的DER编码("certificate verify")

```
30
45
022100
b2e2c924d43aaa5ae11492e34b0c3cb1cfbd4c1742b3eead44a89e7129427a06
0220
652a4550b0bd8f1594c5bc7c8f370539fe6a57d5e1a17e78ff99e0566a1a4a20
```

- 解码得到实际签名值

```
b2e2c924d43aaa5ae11492e34b0c3cb1cfbd4c1742b3eead44a89e7129427a06652a4550b0bd
8f1594c5bc7c8f370539fe6a57d5e1a17e78ff99e0566a1a4a20
```

- 客户端签名公钥

```
04
f1351cf6292f6f4ca4a4b727d81a3285826ce231c169bbb301b552991de5d016fe552ec26aef
8260ca85f2007b9ab836be6a61edf3d9c8567dcce1d1394dcfa9
```

- 客户端签名验证:

签名验签数据

c08d85599e2172797add1a3d510b63f3122815b9716e426facd0ca83a36c60cf

密钥

f1351cf6292f6f4ca4a4b727d81a3285826ce231c169bbb301b552991de5d016fe552ec26aef8260ca85f2007b9ab836be6a61edf3d9c8567dce1d1394dcfa9

签名值

b2e2c924d43aaa5ae11492e34b0c3cb1cfbd4c1742b3eead44a89e7129427a06652a4550b0bd8f1594c5bc7c8f370539fe6a57d5e1a17e78ff99e0566a1a4a20

签名

验签

结果

验证成功

密钥计算

预主密钥验证

- 私钥解析
  - 私钥内容(来自模拟题流量包服务端加密私钥.pem的内容)

-----BEGIN PRIVATE KEY-----  
MIGTAgEAMBMGBYqGSM49AgEGCCqBHM9VAYItBHKwdwIBAQQgb8tU18wJ17FqiOSp  
pHwKa1pOgxQhAP0VoPe/EU0cGAqGcgYIKoEcz1UBgi2hRANCAARipb11eBDL1uL8  
OMWZOfdyOXKsvWMu40pEW2nFdUPxWpRw+akHNz8Aat6Mbr5k9fUiG3gbC/SYG44t  
d40YCWfb  
-----END PRIVATE KEY-----



公钥解析 × 私钥解析 ×

私钥

-----BEGIN PRIVATE KEY-----  
MIGTAqEAMBMGBYqGSM49AgEGCCqBHM9VAYItBHkdwIBAQQgb8tU18wJ17Fqi0Sp  
pHwKa1p0gxQhAP0VoPe/EU0cGAqgCgYIKoEcz1UBgi2hRANCAARipb1leBDL1uL8  
OMWZOfdyOXKsvWMu40pEW2nFdUPxWpRw+akHNz8Aat6Mbr5k9FuiG3gbC/SYG44t  
d40YCWFBh

解析私钥

私钥

6fcb5497cc09d7b16a88e4a9a47c0a6b5a4e83142100fd15a0f7bf114d1c180a

公钥

62a5bd757810cb96e2fc38c59939f7723972acbd632ee34a445b69c57543f15a9470f9a907373f006ade8c6ebe64f5f5221b781b0bf4981b8e2d778d1809615b

◦ 私钥十六进制值

6fcb5497cc09d7b16a88e4a9a47c0a6b5a4e83142100fd15a0f7bf114d1c180a

◦ 公钥十六进制值

62a5bd757810cb96e2fc38c59939f7723972acbd632ee34a445b69c57543f15a9470f9a907373f006ade8c6ebe64f5f5221b781b0bf4981b8e2d778d1809615b

• 预主密钥验证

◦ 打开模拟题SSL流量包.pcapng

◦ 加密的预主密钥的DER编码值(在"Client Key Exchange"找到"Encrypted PreMaster")

30  
8198  
02  
20  
2f1233582f931890b0d6472d39c1729cc20e00d9b26a0ad119c468b8bbbb3e1a  
02  
20  
171274bf57ef071e4d76ee3441f39b31996b2438e5f3334f91533e93aa1c3db9  
04  
20  
a1e8b8721d5f922c6e33fb7052a2afb0038d05017e2c0ab5035bf772da2cc2f3

```
04
30
c2ce0c6634f3d7626db03963d8c19d8a98c1c4756659b707547fdf0439505e4c1bca724
06c22d6ee62e1811d016b402e
```

- 解码得到加密的预主密钥值

```
2f1233582f931890b0d6472d39c1729cc20e00d9b26a0ad119c468b8bbbb3e1a171274b
f57ef071e4d76ee3441f39b31996b2438e5f3334f91533e93aa1c3db9a1e8b8721d5f92
2c6e33fb7052a2afb0038d05017e2c0ab5035bf772da2cc2f3c2ce0c6634f3d7626db03
963d8c19d8a98c1c4756659b707547fdf0439505e4c1bca72406c22d6ee62e1811d016b
402e
```

- 使用工具解密，得到服务端预主密钥明文

×

公钥解析

×

私钥解析

×

加密解密

×

加密/解密数据

2f1233582f931890b0d6472d39c1729cc20e00d9b26a0ad119c468b8bbbb3e1a171274bf57ef071e4d76ee3441f39b31996b2438e5f3334f91533e93aa1c3db9a1e8b8721d5f922c6e33fb7052a2afb0038d05017e2c0ab5035bf772da2cc2f3c2ce0c6634f3d7626db03963d8c19d8a98c1c4756659b707547fdf0439505e4c1bca72406c22d6ee62e1811d016b402e

密钥

6fcb5497cc09d7b16a88e4a9a47c0a6b5a4e83142100fd15a0f7bf114d1c180a

加密解密

结果

0101ac14a3d2da7dccfcaa9fee0882f41b9974aa17b8a3070c042df50a275caaec6ed730dfd132ea871bdd19574dc83f

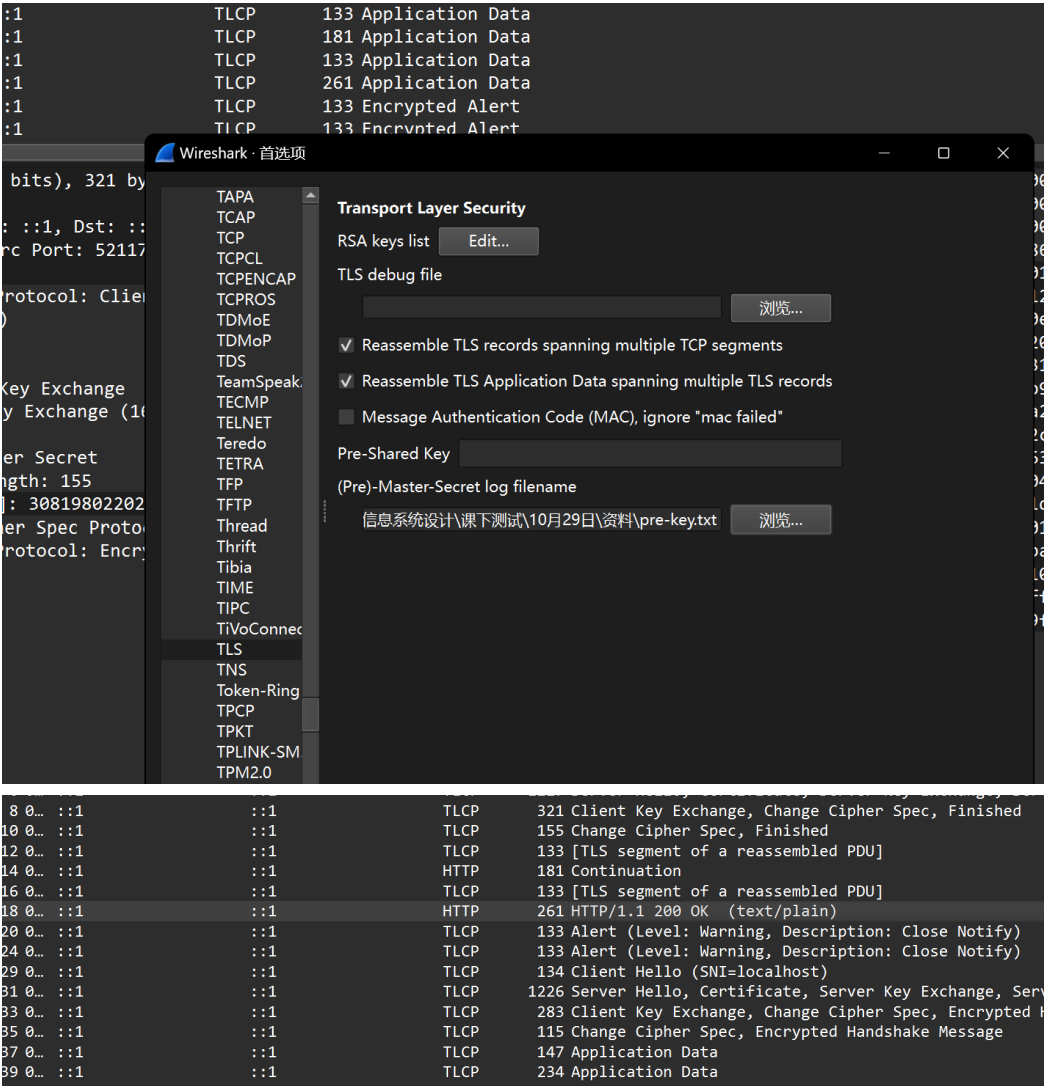
- 预主密钥的明文

```
0101ac14a3d2da7dccfcaa9fee0882f41b9974aa17b8a3070c042df50a275caaec6ed73
0dfd132ea871bdd19574dc83f
```

- 在Wireshark中上传预主密钥明文，可以解密数据包中的密文
  - 建立一个任意文件名、密钥文件内容包括“PMS\_CLIENT\_RANDOM”+client随机数+预主密钥明文”的文本文件
  - 具体内容

PMS\_CLIENT\_RANDOM  
0a359775d8a2c2fd2756261b8ae86767db7b3cd3be532c9665ea432dc260b6d8  
0101ac14a3d2da7dccfcaa9fee0882f41b9974aa17b8a3070c042df50a275caaec  
6ed730dfd132ea871bdd19574dc83f

- 上传该文件
  - 在编辑/首选项/协议/TLS中选中该文件，再点击应用即可看到效果



计算主密钥

- 基本信息
  - 主密钥由48个字节组成，用于生成工作密钥
  - 计算公式： $master\_secret = PRF(pre\_master\_secret, "master secret", ClientHello.random + ServerHello.random)[0..47]$
  - 其中PRF（伪随机函数）的定义是： $PRF(secret, label, seed) = P\_hash(secret, label + seed)$
  - 这里的P\_hash函数是通过  $HMAC(secret, A(1) + seed) + HMAC(secret, A(2) + seed) + HMAC(secret, A(3) + seed) + \dots$  的方式迭代计算的，直到产生所需长度的数据。其中，A(0)被定义为seed，而A(i)是通过  $HMAC(secret, A(i-1))$  计算得到的。
- “mastersecret”的ASCII码

```
root@Youer:~/shiyang# echo -n "master secret" | od -tx1
0000000 6d 61 73 74 65 72 20 73 65 63 72 65 74
0000015
```

- 从"2模拟题SSL流量包"中获取所需的值

- client hello 的random值

```
0a359775d8a2c2fd2756261b8ae86767db7b3cd3be532c9665ea432dc260b6d8
```

- server hello 的random值

```
5423003bf0b79c2a134837d6c560dfb6d11cebdb2ab84e701fafe387b31ac268
```

- 计算seed值

- 公式:  $A0(\text{seed})$  为 "字符串"mastersecret" 的 ASCII 码 || ClientHello.random || ServerHello.random"
- 值:

```
6d6173746572207365637265740a359775d8a2c2fd2756261b8ae86767db7b3cd3be532c9665ea432dc260b6d85423003bf0b79c2a134837d6c560dfb6d11cebdb2ab84e701fafe387b31ac268
```

- $A1$   $A1 = \text{HMAC}(\text{PMS}, A0)$  -- PMS 预主密钥

```
75466ba297baa6abf75fa19698b5debbdf815d632b98fdaf7c1bf60d8a7724f9e
```

- $A2$   $A2 = \text{HMAC}(\text{PMS}, A1)$

```
c7642fabae939cad31063af385648bb48b8fb31e4c94ed97e1cc271208c0e477
```

- $A1 || \text{seed}$

```
75466ba297baa6abf75fa19698b5debbdf815d632b98fdaf7c1bf60d8a7724f9e6d6173746572207365637265740a359775d8a2c2fd2756261b8ae86767db7b3cd3be532c9665ea432dc260b6d85423003bf0b79c2a134837d6c560dfb6d11cebdb2ab84e701fafe387b31ac268
```

- $ms1$   $ms1 = \text{HMAC}(\text{PMS}, A1 || \text{seed})$

```
640da76978a5efd6e0953088f015fa20ee810e48e24aa6f19a8900bcff05a3a7
```

- $A2 || \text{seed}$

```
c7642fabae939cad31063af385648bb48b8fb31e4c94ed97e1cc271208c0e4776d6173746572
207365637265740a359775d8a2c2fd2756261b8ae86767db7b3cd3be532c9665ea432dc260b6
d85423003bf0b79c2a134837d6c560dfb6d11cebdb2ab84e701fafe387b31ac268
```

- $ms2 \quad ms2 = \text{HMAC}(\text{PMS}, A2 || \text{seed})$

```
722de36c807e5c7ca912e7e6ffa4965e7db07bef99df8c0f27f29c21237f94c7
```

- $ms1 || ms2$

```
640da76978a5efd6e0953088f015fa20ee810e48e24aa6f19a8900bcff05a3a7722de36c807e
5c7ca912e7e6ffa4965e7db07bef99df8c0f27f29c21237f94c7
```

- 主密钥  $\text{master\_secret} = (ms1 || ms2)[0...47]$

```
640da76978a5efd6e0953088f015fa20ee810e48e24aa6f19a8900bcff05a3a7722de36c807e
5c7ca912e7e6ffa4965e
```

#### 计算工作密钥

- 基本信息
  - 工作密钥定义：工作密钥包括校验密钥和加密密钥，具体密钥长度由选用的密码算法决定。由主密钥、客户端随机数、服务端随机数、常量字符串，经PRF计算生成
  - 计算方法如下： $\text{key\_block} = \text{PRF}(\text{SecurityParameters.master\_secret}, "key \text{ expansion}", \text{SecurityParameters.server\_random} + \text{SecurityParameters.client\_random})$ ;
  - 直到生成所需长度的输出，然后按顺序分割得到所需的密钥：
    - $\text{client\_write\_MAC\_secret}[\text{SecurityParameters.hash\_size}]$
    - $\text{server\_write\_MAC\_secret}[\text{SecurityParameters.hash\_size}]$
    - $\text{client\_write\_key}[\text{SecurityParameters.key\_material\_length}]$
    - $\text{server\_write\_key}[\text{SecurityParameters.key\_material\_length}]$
    - $\text{client\_write\_IV}[\text{SecurityParameters.fixed\_iv\_length}]$
    - $\text{server\_write\_IV}[\text{SecurityParameters.fixed\_iv\_length}]$
  - 注意这些密钥一共需要1024字节（ $\text{sm3hash\_size} : 256\text{bits}; \text{sm4\_ey\_material\_length} : 128\text{bits}; \text{sm4\_fixed\_iv\_length} : 128\text{bits}$ ），需要四次HMAC计算，下面分别记为 $A1, A2, A3, A4$
- 计算过程

- "key expansion"ASCII码

```
root@Youer:~# echo -n "key expansion" | od -tc -tx1
0000000  k  e  y           e  x  p  a  n  s  i  o  n
          6b 65 79 20 65 78 70 61 6e 73 69 6f 6e
0000015
```

- server\_random

```
5423003bf0b79c2a134837d6c560dfb6d11cebdb2ab84e701fafe387b31ac268
```

- client\_random

```
0a359775d8a2c2fd2756261b8ae86767db7b3cd3be532c9665ea432dc260b6d8
```

- A0(seed) : "keyexpansion"||server\_random||client\_random

```
6b657920657870616e73696f6e5423003bf0b79c2a134837d6c560dfb6d11cebdb2ab84
e701fafe387b31ac2680a359775d8a2c2fd2756261b8ae86767db7b3cd3be532c9665ea
432dc260b6d8
```

- A1: A1=HMAC(MS,A0)

```
6003a5937faa6c9276656bd9cf11e5fc44d1da8cf8916cc3f1a1d9c9c5b2c364
```

- A1||seed

```
6003a5937faa6c9276656bd9cf11e5fc44d1da8cf8916cc3f1a1d9c9c5b2c3646b65792
0657870616e73696f6e5423003bf0b79c2a134837d6c560dfb6d11cebdb2ab84e701faf
e387b31ac2680a359775d8a2c2fd2756261b8ae86767db7b3cd3be532c9665ea432dc26
0b6d8
```

- kb1: HMAC(MS,A1||seed)

```
87e00935dc21b9d87e3e5616179307d817966079a0dea9b9d09a4883cbf0e13d
```

- A2: HMAC(MS,A1)

```
0a45328c9d23a88ad0b0dfe3656996b979bd4a2730a3f8ede29bc392153b6a6b
```

- A2||seed

```
0a45328c9d23a88ad0b0dfe3656996b979bd4a2730a3f8ede29bc392153b6a6b6b65792
0657870616e73696f6e5423003bf0b79c2a134837d6c560dfb6d11cebdb2ab84e701faf
e387b31ac2680a359775d8a2c2fd2756261b8ae86767db7b3cd3be532c9665ea432dc26
0b6d8
```

- kb2=HMAC(MS,A2||seed)

```
b4b53c3b2acafe5781712816c1a3a2cece09d5837eb3389bdf1e471b06bc8157
```

- A3=HMAC(MS,A2)

```
fee4db5e22a5e48c3c0e0b88509763940a8e65c4aaf850c00ae7eb72282089cf
```

- A3||seed

```
fee4db5e22a5e48c3c0e0b88509763940a8e65c4aaf850c00ae7eb72282089cf6b65792
0657870616e73696f6e5423003bf0b79c2a134837d6c560dfb6d11cebdb2ab84e701faf
e387b31ac2680a359775d8a2c2fd2756261b8ae86767db7b3cd3be532c9665ea432dc26
0b6d8
```

- kb3: HMAC(MS,A3||seed)

```
022a23065301de9e49aaa0879e4ecbf2622cda5b8994a0bb4496f03b8c13a4cd
```

- A4: HMAC(MS,A3)

```
8807466c968862f58c17b171b3520ab2a519d5df00c8fc991e9d696244c95437
```

- A4||seed

```
8807466c968862f58c17b171b3520ab2a519d5df00c8fc991e9d696244c954376b65792
0657870616e73696f6e5423003bf0b79c2a134837d6c560dfb6d11cebdb2ab84e701faf
e387b31ac2680a359775d8a2c2fd2756261b8ae86767db7b3cd3be532c9665ea432dc26
0b6d8
```

- kb4: HMAC(MS,A4||seed)

```
145ba093616e5939aae6b59e4bed770145482c297f4c7cc76b497fcb1b2f4a84
```

- 工作密钥kb1||kb2||kb3||kb4

```
87e00935dc21b9d87e3e5616179307d817966079a0dea9b9d09a4883cbf0e13db4b53c3
b2acafe5781712816c1a3a2cece09d5837eb3389bdf1e471b06bc8157022a23065301de
9e49aaa0879e4ecbf2622cda5b8994a0bb4496f03b8c13a4cd145ba093616e5939aae6b
59e4bed770145482c297f4c7cc76b497fcb1b2f4a84
```

- 由MAC256bits,key128bits,IV128bits规则分解得到所需的各种密钥

```
client MAC:
87e00935dc21b9d87e3e5616179307d817966079a0dea9b9d09a4883cbf0e13d
server MAC:
b4b53c3b2acafe5781712816c1a3a2cece09d5837eb3389bdf1e471b06bc8157
client write key: 022a23065301de9e49aaa0879e4ecbf2
server write key: 622cda5b8994a0bb4496f03b8c13a4cd
client IV: 145ba093616e5939aae6b59e4bed7701
server IV: 45482c297f4c7cc76b497fcb1b2f4a84
```

## 使用OpenSSL或者GmSSL命令验证

- 使用GmSSL命令验证签名是否正确
  - 验证客户端证书签名
    - 通过[在线网站](#)将公钥由HEX格式转换成PEM格式

```
-----BEGIN PUBLIC KEY-----
MFkwEwYHKoZIzj0CAQYIKoEcz1UBgi0DQgAE8TUc9ikvb0ykpLcn2BoyhYJs4jHB
abuzAbVSmR3l0Bb+VS7Cau+CYMqF8gB7mrg2vmph7fPZyFZ9zOHROU3PqQ==
-----END PUBLIC KEY-----
```

- 创建文件（这里的签名值用的是真实值，不是DER编码值，之后会改为DER编码值）

```
root@Youer:~/TestInClass/test1029/client-sign# nano public_key.pem
root@Youer:~/TestInClass/test1029/client-sign# echo -n
"c08d85599e2172797add1a3d510b63f3122815b9716e426facd0ca83a36c60cf"
| xxd -r -p > message.txt
root@Youer:~/TestInClass/test1029/client-sign# echo
"b2e2c924d43aaa5ae11492e34b0c3cb1cfbd4c1742b3eead44a89e7129427a066
```



```
52a4550b0bd8f1594c5bc7c8f370539fe6a57d5e1a17e78ff99e0566a1a4a20" |
xxd -r -p > signature.bin
```

## ■ 查询命令

```
root@Youer:~/TestInClass/test1029/client-sign# gmssl sm2verify -
help
usage: gmssl sm2verify (-pubkey pem | -cert pem) [-id str] [-in
file] -sig file
```

### Options

-pubkey pem	Signer's public key file in PEM format
-cert pem	Signer's certificate in PEM format
-id str	Signer's identity string,
'1234567812345678'	by default
-in file   stdin	Signed file or data
-sig file	Signature in binary DER encoding

### Examples

```
$ gmssl sm2keygen -pass P@ssw0rd -out sm2.pem -pubout
sm2pub.pem
$ echo -n 'message to be signed' | gmssl sm2sign -key sm2.pem
-pass P@ssw0rd -out sm2.sig
$ echo -n 'message to be signed' | gmssl sm2verify -pubkey
sm2pub.pem -sig sm2.sig
```

## ■ 验证过程（中途将签名值改为DER编码的值而不是真实值）

```
root@Youer:~/TestInClass/test1029/client-sign# ls
message.txt public_key.pem signature.bin
root@Youer:~/TestInClass/test1029/client-sign# gmssl sm2verify -
pubkey public_key.pem -in message.txt -sig signature.bin
/root/GmSSL/src/sm2_sign.c:664:sm2_verify_finish():
gmssl sm2verify: inner error
root@Youer:~/TestInClass/test1029/client-sign# ls
message.txt public_key.pem signature.bin
root@Youer:~/TestInClass/test1029/client-sign# echo -n
"3045022100b2e2c924d43aaa5ae11492e34b0c3cb1cfbd4c1742b3eead44a89e7
129427a060220652a4550b0bd8f1594c5bc7c8f370539fe6a57d5e1a17e78ff99e
0566a1a4a20" | xxd -r -p > signature.bin
root@Youer:~/TestInClass/test1029/client-sign# gmssl sm2verify -
pubkey public_key.pem -in message.txt -sig signature.bin
verify : success
```

- 验证服务端证书签名

- 通过[在线网站](#)将公钥由HEX格式转换成PEM格式

```
-----BEGIN PUBLIC KEY-----
MFkwEwYHKoZIzj0CAQYIKoEcz1UBgi0DQgAEEnNbxed9qX9K3Hyp3J2S0/FRoHIRE
KFQhvc34Hu5Lqj2V5QsEpZIHhEY2h4KrJJXK+p1DbPX6ul2Xr/acRD8Tn1w==
-----END PUBLIC KEY-----
```

- 验证过程

```
root@Youer:~/TestInClass/test1029/server-sign# nano public_key.pem
root@Youer:~/TestInClass/test1029/server-sign# echo -n
"6541f880b28ee1eaea9d5a92679ce16cea85c56df7ede810ffa78d44fb0b944d5
53c83b288b79abae1841fe1e05ff29d0771d3160314d562444f574e47524400000
1f7308201f330820198a00302010202145326ccffbb03ba78b831599da3e085133
de83da8300a06082a811ccf550183753068310b300906035504061302434e31113
00f06035504080c085368616e646f6e67310f300d060355040a0c0653616e73656
33122302006092a864886f70d01090116136e656967686261647340676d61696c2
e636f6d3111300f06035504030c08736d325f726f6f74301e170d3230303730323
036303030355a170d3330303633303036303030355a306e310b300906035504061
302434e3111300f06035504080c085368616e646f6e67310f300d060355040a0c0
653616e7365633122302006092a864886f70d01090116136e65696768626164734
0676d61696c2e636f6d3117301506035504030c0e736d325f7365727665725f656
e633059301306072a8648ce3d020106082a811ccf5501822d0342000499fe11af6
3e5cc352759771030cf5ab549f4f2ee3ccd8126345d81d0116aa15087977e93037
01cc565ecae91441b670af288eade4c579ef817f2ad65d054fda2a31a301830090
603551d1304023000300b0603551d0f040403020470300a06082a811ccf5501837
50349003046022100fb2cfe11bdacf92a9f2b10bdfdb9e6d7fff08e8683ee63b440
b25c5c6836f85c5022100b66a71e1236de75383976a5e8d38f06f0db145ae01ea8
313231edfe35f215639" | xxd -r -p > message.txt
root@Youer:~/TestInClass/test1029/server-sign# echo
"3046022100d111933a5d9b42e7010e8787dcc6c28eda4e853fde9b0a27f8aa593
7d091afdc0221008dd88b5116738935cefad8b5cd7a5f7b554ba92cd20d22e722a
055f4fc2e4d61" | xxd -r -p > signature.bin
root@Youer:~/TestInClass/test1029/server-sign# gmssl sm2verify -
pubkey public_key.pem -in message.txt -sig signature.bin
verify : success
root@Youer:~/TestInClass/test1029/server-sign#
```

- 使用openssl命令进行私钥解析过程

```
root@Youer:~/TestInClass/test1029/pri_key_jiexi# nano sm2private_key.pem
root@Youer:~/TestInClass/test1029/pri_key_jiexi# openssl asn1parse -inform
PEM -in sm2private_key.pem
 0:d=0 hl=3 l= 147 cons: SEQUENCE
 3:d=1 hl=2 l=  1 prim: INTEGER           :00
 6:d=1 hl=2 l= 19 cons: SEQUENCE
 8:d=2 hl=2 l=  7 prim: OBJECT           :id-ecPublicKey
```

```
17:d=2 hl=2 l= 8 prim: OBJECT :sm2
27:d=1 hl=2 l= 121 prim: OCTET STRING [HEX
DUMP]:307702010104206FCB5497CC09D7B16A88E4A9A47C0A6B5A4E83142100FD15A0F7BF11
4D1C180AA00A06082A811CCF5501822DA1440342000462A5BD757810CB96E2FC38C59939F772
3972ACBD632EE34A445B69C57543F15A9470F9A907373F006ADE8C6EBE64F5F5221B781B0BF4
981B8E2D778D1809615B
root@Youer:~/TestInClass/test1029/pri_key_jiexi# openssl base64 -d -in
sm2private_key.pem -out sm2private_key.der
root@Youer:~/TestInClass/test1029/pri_key_jiexi# openssl asn1parse -inform
DER -in sm2private_key.der
    0:d=0 hl=3 l= 147 cons: SEQUENCE
      3:d=1 hl=2 l= 1 prim: INTEGER :00
      6:d=1 hl=2 l= 19 cons: SEQUENCE
        8:d=2 hl=2 l= 7 prim: OBJECT :id-ecPublicKey
17:d=2 hl=2 l= 8 prim: OBJECT :sm2
27:d=1 hl=2 l= 121 prim: OCTET STRING [HEX
DUMP]:307702010104206FCB5497CC09D7B16A88E4A9A47C0A6B5A4E83142100FD15A0F7BF11
4D1C180AA00A06082A811CCF5501822DA1440342000462A5BD757810CB96E2FC38C59939F772
3972ACBD632EE34A445B69C57543F15A9470F9A907373F006ADE8C6EBE64F5F5221B781B0BF4
981B8E2D778D1809615B
root@Youer:~/TestInClass/test1029/pri_key_jiexi# od -tx1 sm2private_key.der
0000000 30 81 93 02 01 00 30 13 06 07 2a 86 48 ce 3d 02
0000020 01 06 08 2a 81 1c cf 55 01 82 2d 04 79 30 77 02
0000040 01 01 04 20 6f cb 54 97 cc 09 d7 b1 6a 88 e4 a9
0000060 a4 7c 0a 6b 5a 4e 83 14 21 00 fd 15 a0 f7 bf 11
0000100 4d 1c 18 0a a0 0a 06 08 2a 81 1c cf 55 01 82 2d
0000120 a1 44 03 42 00 04 62 a5 bd 75 78 10 cb 96 e2 fc
0000140 38 c5 99 39 f7 72 39 72 ac bd 63 2e e3 4a 44 5b
0000160 69 c5 75 43 f1 5a 94 70 f9 a9 07 37 3f 00 6a de
0000200 8c 6e be 64 f5 f5 22 1b 78 1b 0b f4 98 1b 8e 2d
0000220 77 8d 18 09 61 5b
0000226
root@Youer:~/TestInClass/test1029/pri_key_jiexi# openssl pkey -in
sm2private_key.pem -text -noout
Private-Key: (256 bit)
priv:
    6f:cb:54:97:cc:09:d7:b1:6a:88:e4:a9:a4:7c:0a:
    6b:5a:4e:83:14:21:00:fd:15:a0:f7:bf:11:4d:1c:
    18:0a
pub:
    04:62:a5:bd:75:78:10:cb:96:e2:fc:38:c5:99:39:
    f7:72:39:72:ac:bd:63:2e:e3:4a:44:5b:69:c5:75:
    43:f1:5a:94:70:f9:a9:07:37:3f:00:6a:de:8c:6e:
    be:64:f5:f5:22:1b:78:1b:0b:f4:98:1b:8e:2d:77:
    8d:18:09:61:5b
ASN1 OID: SM2
```