

哈尔滨工业大学(深圳)

《网络与系统安全》 实 验报告

实验五

TLS 实验

学 院: 计算机科学与技术
姓 名: 宗晴
学 号: 200110513
专 业: 计算机
日 期: 2023 年 5 月

1.在客户端容器中执行如下命令 `./handshake.py www.baidu.com` 根据执行结果回答下面三个问题。

```

root@743cc08b925f:/volumes# ./handshake.py www.baidu.com
After making TCP connection. Press any key to continue ...
=== Cipher used: ('ECDHE-RSA-AES128-GCM-SHA256', 'TLSv1.2', 128)
=== Server hostname: www.baidu.com
=== Server certificate:
{'OCSP': ('http://ocsp.globalsign.com/gsrsoavsslca2018',),
'caIssuers': ('http://secure.globalsign.com/cacert/gsrsoavsslca2018.crt',),
'crlDistributionPoints': ('http://crl.globalsign.com/gsrsoavsslca2018.crl',),
'issuer': (((('countryName', 'BE'),),
              (('organizationName', 'GlobalSign nv-sa'),),
              (('commonName', 'GlobalSign RSA OV SSL CA 2018'),)),
'notAfter': 'Aug  6 05:16:01 2023 GMT',
'notBefore': 'Jul  5 05:16:02 2022 GMT',
'serialNumber': '4417CE86EF82EC6921CC6F68',
'subject': (((('countryName', 'CN'),),
              (('stateOrProvinceName', 'beijing'),),
              (('localityName', 'beijing'),),
              (('organizationalUnitName', 'service operation department'),),
              (('organizationName',
                'Beijing Baidu Netcom Science Technology Co., Ltd'),),
              (('commonName', 'baidu.com'),)),
'subjectAltName': (('DNS', 'baidu.com'),
                  ('DNS', 'baifubao.com'),
                  ('DNS', 'www.baidu.cn'),
                  ('DNS', 'www.baidu.com.cn'),
                  ('DNS', 'mct.y.nuomi.com'),
                  ('DNS', 'apollo.auto'),

                  ('DNS', 'mct.y.nuomi.com'),
                  ('DNS', 'apollo.auto'),
                  ('DNS', 'dwz.cn'),
                  ('DNS', '*.baidu.com'),
                  ('DNS', '*.baifubao.com'),
                  ('DNS', '*.baidustatic.com'),
                  ('DNS', '*.bdstatic.com'),
                  ('DNS', '*.bdimg.com'),
                  ('DNS', '*.hao123.com'),
                  ('DNS', '*.nuomi.com'),
                  ('DNS', '*.chuanke.com'),
                  ('DNS', '*.trustgo.com'),
                  ('DNS', '*.bce.baidu.com'),
                  ('DNS', '*.eyun.baidu.com'),
                  ('DNS', '*.map.baidu.com'),
                  ('DNS', '*.mbd.baidu.com'),
                  ('DNS', '*.fanyi.baidu.com'),
                  ('DNS', '*.baidubce.com'),
                  ('DNS', '*.mipcdn.com'),
                  ('DNS', '*.news.baidu.com'),
                  ('DNS', '*.baidupcs.com'),
                  ('DNS', '*.aipage.com'),
                  ('DNS', '*.aipage.cn'),
                  ('DNS', '*.bcehost.com'),
                  ('DNS', '*.safe.baidu.com'),
                  ('DNS', '*.im.baidu.com'),
                  ('DNS', '*.baiducontent.com'),
                  ('DNS', '*.dlnel.com'),
                  ('DNS', '*.dlnel.org'),

```

```

        ('DNS', '*.su.baidu.com'),
        ('DNS', '*.91.com'),
        ('DNS', '*.hao123.baidu.com'),
        ('DNS', '*.apollo.auto'),
        ('DNS', '*.xueshu.baidu.com'),
        ('DNS', '*.bj.baidubce.com'),
        ('DNS', '*.gz.baidubce.com'),
        ('DNS', '*.smartapps.cn'),
        ('DNS', '*.bdtjrcv.com'),
        ('DNS', '*.hao222.com'),
        ('DNS', '*.haokan.com'),
        ('DNS', '*.pae.baidu.com'),
        ('DNS', '*.vd.bdstatic.com'),
        ('DNS', '*.cloud.baidu.com'),
        ('DNS', 'click.hm.baidu.com'),
        ('DNS', 'log.hm.baidu.com'),
        ('DNS', 'cm.pos.baidu.com'),
        ('DNS', 'wn.pos.baidu.com'),
        ('DNS', 'update.pan.baidu.com')),
    'version': 3}
[{'issuer': (((('organizationalUnitName', 'GlobalSign Root CA - R3'),),
                (('organizationName', 'GlobalSign'),),
                (('commonName', 'GlobalSign'),))),
  'notAfter': 'Mar 18 10:00:00 2029 GMT',
  'notBefore': 'Mar 18 10:00:00 2009 GMT',
  'serialNumber': '04000000000121585308A2',
  'subject': (((('organizationalUnitName', 'GlobalSign Root CA - R3'),),
                 (('organizationName', 'GlobalSign'),),
                 (('commonName', 'GlobalSign'),))),
  'version': 3}]
After TLS handshake. Press any key to continue ...
root@743cc08b925f:/volumes# █

```

(1) 客户端和服务端使用的加密算法有哪些，分别起什么作用？

客户端和服务端使用的加密算法有：
ECDHE-RSA-AES128-GCM-SHA256。

RSA 是一种公钥密码算法，是第一个能同时用于加密和数字签名的算法，也易于理解和操作。

ECDHE 算法是在 DHE 算法的基础上利用了 ECC 椭圆曲线特性，可以用更少的计算量计算出公钥，以及最终的会话密钥。具有前向安全，被广泛使用。

AES128 是一个分组密码，属于对称密码范畴，AES 算法的模块在对称密码领域特别是分组密码领域常有使用。AES128 分组长度为 128 比特，密

钥长度也为 128 比特。算法涉及 4 种操作：字节替代、行移位、列混淆和轮密钥加。

GCM (Galois/Counter Mode) 指的是该对称加密采用 Counter 模式, 并带有 GMAC 消息认证码。二者分别保证了加密算法的保密性、完整性。

SHA-256 作为 SHA-2(Secure Hash Algorithm 2, 安全哈希算法 2) 的一部分, 目前已经是最流行的哈希算法之一。安全加密算法通过将输入文本拆分成独立的片段, 并通过这些独立的片段生成最终的结果——加密算法哈希值。这些加密算法哈希值几乎是唯一的字符串, 因而它们往往被用作数据块的摘要"digest", 指纹"fingerprint"或签名"signature"。SHA-256 算法往往被用来生成 256 位的签名。

(2) 分析打印出来的服务器端证书

服务器端证书如上图所示, 为"===Server certificate: "字段后的内容。

在打印出来的服务器端证书前首先说明了签名算法、主机名(www.baidu.com)。然后证书中说明了所用的 OCSP 即在线证书状态协议、calssuers 即证书的签发机构等信息, 以及 notAfter 和 notBefore 即证书的有效时间、serialNumber 即证书的独特序列号、subject 即证书的拥有者信息、subjectAltName 即证书拥有者的别名、version 即版本号为 3, 以及其他相关信息。

(3) 抓包分析 TLS 握手协议

抓包结果如下:

| Apply a display filter ... <Ctrl-/> | | | | | | |
|---|-------------------------------|-------------------------|-----------------|----------|--------|---|
| No. | Time | Source | Destination | Protocol | Length | Info |
| 1 | 2023-05-25 04:30:22.122884276 | 10.0.2.15 | 10.248.98.30 | DNS | 73 | Standard query 0xc97d A www.baidu.com |
| 2 | 2023-05-25 04:30:22.126072922 | 10.248.98.30 | 10.0.2.15 | DNS | 132 | Standard query response 0xc97d A www.baidu.com CNAME www.a.shifen.com A 14.119.104.189 A 14.119.104.254 |
| 3 | 2023-05-25 04:30:22.126038536 | 10.0.2.15 | 14.119.104.189 | TCP | 74 | 37016 → 443 [SYN] Seq=2408693201 Win=0 MSS=1460 SACK_PERM=1 TSval=81460810 TSecr=0 WS=128 |
| 4 | 2023-05-25 04:30:22.136877329 | 14.119.104.189 | 10.0.2.15 | TCP | 60 | 443 → 37016 [SYN, ACK] Seq=171072001 Ack=2408693202 Win=65535 Len=0 MSS=1460 |
| 5 | 2023-05-25 04:30:22.136911919 | 10.0.2.15 | 14.119.104.189 | TCP | 54 | 37016 → 443 [ACK] Seq=2408693202 Ack=171072002 Win=64240 Len=0 |
| 6 | 2023-05-25 04:30:23.107664666 | 10.0.2.15 | 14.119.104.189 | TLSv1.2 | 571 | Client Hello |
| 7 | 2023-05-25 04:30:23.107949633 | 14.119.104.189 | 10.0.2.15 | TCP | 60 | 443 → 37016 [ACK] Seq=171072002 Ack=2408693719 Win=65535 Len=0 |
| 8 | 2023-05-25 04:30:23.125658230 | 14.119.104.189 | 10.0.2.15 | TLSv1.2 | 1506 | Server Hello |
| 9 | 2023-05-25 04:30:23.125705069 | 10.0.2.15 | 14.119.104.189 | TCP | 54 | 37016 → 443 [ACK] Seq=2408693719 Ack=171073454 Win=63888 Len=0 |
| 10 | 2023-05-25 04:30:23.127974793 | 14.119.104.189 | 10.0.2.15 | TLSv1.2 | 3828 | Certificate, Server Key Exchange, Server Hello Done |
| 11 | 2023-05-25 04:30:23.127997348 | 10.0.2.15 | 14.119.104.189 | TCP | 54 | 37016 → 443 [ACK] Seq=2408693719 Ack=171077228 Win=61320 Len=0 |
| 12 | 2023-05-25 04:30:23.128720708 | 10.0.2.15 | 14.119.104.189 | TLSv1.2 | 180 | Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message |
| 13 | 2023-05-25 04:30:23.128897947 | 14.119.104.189 | 10.0.2.15 | TCP | 60 | 443 → 37016 [ACK] Seq=171077228 Ack=2408693845 Win=65535 Len=0 |
| 14 | 2023-05-25 04:30:23.139487998 | 14.119.104.189 | 10.0.2.15 | TLSv1.2 | 280 | New Session Ticket, Change Cipher Spec, Encrypted Handshake Message |
| 15 | 2023-05-25 04:30:23.139541951 | 10.0.2.15 | 14.119.104.189 | TCP | 54 | 37016 → 443 [ACK] Seq=2408693845 Ack=171077454 Win=62780 Len=0 |
| 16 | 2023-05-25 04:30:23.937562025 | 10.0.2.15 | 14.119.104.189 | TCP | 54 | 37016 → 443 [FIN, ACK] Seq=2408693845 Ack=171077454 Win=62780 Len=0 |
| 17 | 2023-05-25 04:30:23.937829557 | 14.119.104.189 | 10.0.2.15 | TCP | 60 | 443 → 37016 [ACK] Seq=171077454 Ack=2408693846 Win=65535 Len=0 |
| 18 | 2023-05-25 04:30:23.948125218 | 14.119.104.189 | 10.0.2.15 | TLSv1.2 | 85 | Encrypted Alert |
| 19 | 2023-05-25 04:30:23.948125274 | 14.119.104.189 | 10.0.2.15 | TCP | 60 | 443 → 37016 [FIN, ACK] Seq=171077485 Ack=2408693846 Win=65535 Len=0 |
| 20 | 2023-05-25 04:30:23.949174024 | 10.0.2.15 | 14.119.104.189 | TCP | 54 | 37016 → 443 [RST] Seq=2408693846 Win=0 Len=0 |
| 21 | 2023-05-25 04:30:23.948185055 | 10.0.2.15 | 14.119.104.189 | TCP | 54 | 37016 → 443 [RST] Seq=2408693846 Win=0 Len=0 |
| 22 | 2023-05-25 04:30:23.948406346 | 14.119.104.189 | 10.0.2.15 | TCP | 60 | 443 → 37016 [RST, ACK] Seq=0 Ack=2408693846 Win=0 Len=0 |
| 23 | 2023-05-25 04:30:26.179998106 | 10.0.2.15 | 10.248.98.30 | DNS | 73 | Standard query 0x9a57 A www.baidu.com |
| 24 | 2023-05-25 04:30:26.181555349 | 10.248.98.30 | 10.0.2.15 | DNS | 132 | Standard query response 0x9a57 A www.baidu.com CNAME www.a.shifen.com A 14.119.104.254 A 14.119.104.189 |
| 25 | 2023-05-25 04:30:26.181749858 | 10.0.2.15 | 14.119.104.254 | TCP | 74 | 60118 → 443 [SYN] Seq=2279105157 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=2078778273 TSecr=0 WS=128 |
| 26 | 2023-05-25 04:30:26.182579308 | 14.119.104.254 | 10.0.2.15 | TCP | 60 | 443 → 60118 [SYN, ACK] Seq=171072002 Ack=2279105157 Win=65535 Len=0 MSS=1460 |
| Frame 1: 73 bytes on wire (584 bits), 73 bytes captured (584 bits) on interface enp0s3, id 0 | | | | | | |
| Ethernet II, Src: PcsCompu_9d:97:50 (08:00:27:9d:97:50), Dst: RealtekU_12:35:02 (52:54:00:12:35:02) | | | | | | |
| Internet Protocol Version 4, Src: 10.0.2.15, Dst: 10.248.98.30 | | | | | | |
| Hypertext Transfer Protocol, Request: GET / HTTP/1.1 | | | | | | |
| 0000 | 52 54 00 12 35 02 08 00 | 27 9d 97 50 08 00 45 00 | RT-5-...P-E | | | |
| 0010 | 00 3b 17 b1 40 00 3f 11 | aa dc 0a 00 02 0f 0a f8 | ;...@?... .. | | | |
| 0020 | 62 1e 99 b7 00 35 00 27 | 79 5d c9 7d 01 00 00 01 | b...5...y... .. | | | |
| 0030 | 00 00 00 00 00 00 03 77 | 77 77 05 62 61 69 64 75 |w ww baidu | | | |
| 0040 | 03 63 6f 6d 00 00 01 00 | 01 | .com... | | | |

| tts | | | | | | |
|--|-------------------------------|-------------------------|---------------------|----------|--------|--|
| No. | Time | Source | Destination | Protocol | Length | Info |
| 6 | 2023-05-25 04:30:23.107664666 | 10.0.2.15 | 14.119.104.189 | TLSv1.2 | 571 | Client Hello |
| 8 | 2023-05-25 04:30:23.125658230 | 14.119.104.189 | 10.0.2.15 | TLSv1.2 | 1506 | Server Hello |
| 10 | 2023-05-25 04:30:23.127974793 | 14.119.104.189 | 10.0.2.15 | TLSv1.2 | 3828 | Certificate, Server Key Exchange, Server Hello Done |
| 12 | 2023-05-25 04:30:23.128720708 | 10.0.2.15 | 14.119.104.189 | TLSv1.2 | 180 | Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message |
| 14 | 2023-05-25 04:30:23.139487998 | 14.119.104.189 | 10.0.2.15 | TLSv1.2 | 280 | New Session Ticket, Change Cipher Spec, Encrypted Handshake Message |
| 18 | 2023-05-25 04:30:23.948125218 | 14.119.104.189 | 10.0.2.15 | TLSv1.2 | 85 | Encrypted Alert |
| 28 | 2023-05-25 04:30:27.427639433 | 10.0.2.15 | 14.119.104.254 | TLSv1.2 | 571 | Client Hello |
| 30 | 2023-05-25 04:30:27.444120771 | 14.119.104.254 | 10.0.2.15 | TLSv1.2 | 1506 | Server Hello |
| 32 | 2023-05-25 04:30:27.444394524 | 14.119.104.254 | 10.0.2.15 | TLSv1.2 | 3828 | Certificate, Server Key Exchange, Server Hello Done |
| 34 | 2023-05-25 04:30:27.445325991 | 10.0.2.15 | 14.119.104.254 | TLSv1.2 | 180 | Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message |
| 36 | 2023-05-25 04:30:27.456724613 | 14.119.104.254 | 10.0.2.15 | TLSv1.2 | 280 | New Session Ticket, Change Cipher Spec, Encrypted Handshake Message |
| 40 | 2023-05-25 04:30:28.251682328 | 14.119.104.254 | 10.0.2.15 | TLSv1.2 | 85 | Encrypted Alert |
| Frame 6: 571 bytes on wire (4568 bits), 571 bytes captured (4568 bits) on interface enp0s3, id 0 | | | | | | |
| Ethernet II, Src: PcsCompu_9d:97:50 (08:00:27:9d:97:50), Dst: RealtekU_12:35:02 (52:54:00:12:35:02) | | | | | | |
| Internet Protocol Version 4, Src: 10.0.2.15, Dst: 14.119.104.189 | | | | | | |
| Transmission Control Protocol, Src Port: 37016, Dst Port: 443, Seq: 2408693202, Ack: 171072002, Len: 517 | | | | | | |
| Transport Layer Security | | | | | | |
| 0000 | 52 54 00 12 35 02 08 00 | 27 9d 97 50 08 00 45 00 | RT-5-...P-E | | | |
| 0010 | 02 2d 0e c9 40 00 3f 06 | a7 bf 0a 00 02 0f 0e 77 | ...@?...w | | | |
| 0020 | 68 bd 90 98 01 bb 8f 91 | bd d2 0a 32 5a 02 50 18 | h.....2Z:P | | | |
| 0030 | fa f0 85 62 00 00 16 03 | 01 02 00 01 00 01 fc 03 | ...b..... | | | |
| 0040 | 03 f4 27 2a a8 af c3 32 | f5 ed 1a 04 0d b0 bd dc | ...2..... | | | |
| 0050 | 77 03 26 ce 34 0d 68 43 | 47 2a 0c 52 07 1e 93 1f | w&4hC G'R... | | | |
| 0060 | c7 20 43 a9 c0 81 46 e9 | 8c 82 95 67 b0 08 af ce | ...C...F...g... | | | |
| 0070 | 49 f1 43 d7 34 f0 3c c9 | ca 65 ff 3c 87 36 da b9 | I C 4<...e<6... | | | |
| 0080 | 69 e5 00 3e 13 02 13 03 | 13 01 c0 2c c0 30 00 9f | ...<...>...0... | | | |
| 0090 | cc a9 cc a8 cc aa c0 2b | c0 2f 00 9e c0 24 c0 28 |+.../...\$ (| | | |
| 00a0 | 00 6b c0 23 c0 27 00 67 | c0 0a c0 14 00 39 c0 09 | ...k...#...g...9... | | | |
| 00b0 | c0 13 00 33 00 9d 00 9c | 00 3d 00 3c 00 35 00 2f | ...3.....=<5-/ | | | |

如上图所示，为进行了两次 TLS 握手的抓包过程（已经筛选出了所有的 TLS 数据包）。可以看出 TLS 握手过程如下：

- （1）客户端：发送客户端问候消息（Client Hello），表明它自己支持哪些密码套件和客户端的一次性随机数（Client_random）。
- （2）服务器：发送服务器问候消息（Server Hello），根据客户端发来的问候消息，选择确认一个客户端和服务器都支持的密码套件，并提供服务器的一次性随机数（Server_random）。
- （3）服务器：发送公钥证书（certificate，server key exchange）给客户端。

(4) 服务器：发送握手完成的消息，表明已完成握手协商。(server hello done)

(5) 客户端：发送客户端密钥交换消息 (client key exchange)。客户端随机生成一个预主密钥，然后用服务器的公钥对其进行加密，并将加密后的密钥发送给服务器。客户端和服务端首先使用预主密钥生成主密钥，然后再使用主密钥生成会话密钥。

(6) 客户端和服务端：互相发送更改密码规范消息。

(7) 客户端和服务端：互相发送一个加密完成的消息。

2.更改证书文件路径，请同学们将 www.baidu.com 网站的测试过程截图保存（如果不将证书拷贝过来应该有报错信息，拷贝过来之后应该正常），也可选用其他网站做测试。

如果不将证书拷贝过来，执行 `./handshake.py www.baidu.com` 会有报错信息，如下图所示：

```
root@743cc08b925f:/volumes# ./handshake.py www.baidu.com
After making TCP connection. Press any key to continue ...
Traceback (most recent call last):
  File "./handshake.py", line 29, in <module>
    ssock.do_handshake() # Start the handshake
  File "/usr/lib/python3.8/ssl.py", line 1309, in do_handshake
    self._sslobj.do_handshake()
ssl.SSLCertVerificationError: [SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: unable to get local issuer certificate (_ssl.c:1123)
root@743cc08b925f:/volumes#
root@743cc08b925f:/volumes# ./handshake.py www.baidu.com
After making TCP connection. Press any key to continue ...
Traceback (most recent call last):
  File "./handshake.py", line 29, in <module>
    ssock.do_handshake() # Start the handshake
  File "/usr/lib/python3.8/ssl.py", line 1309, in do_handshake
    self._sslobj.do_handshake()
ssl.SSLCertVerificationError: [SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: unable to get local issuer certificate (_ssl.c:1123)
root@743cc08b925f:/volumes# █
```

如果将证书拷贝过来之后，显示正常，如下图所示：

```
root@743cc08b925f:/volumes# ./handshake.py www.baidu.com
After making TCP connection. Press any key to continue ...
=== Cipher used: ('ECDHE-RSA-AES128-GCM-SHA256', 'TLSv1.2', 128)
=== Server hostname: www.baidu.com
=== Server certificate:
{'OCSP': ('http://ocsp.globalsign.com/gsrsoovsslca2018',),
 'caIssuers': ('http://secure.globalsign.com/cacert/gsrsoovsslca2018.crt',),
 'crlDistributionPoints': ('http://crl.globalsign.com/gsrsoovsslca2018.crl',),
 'issuer': (((('countryName', 'BE'),),
               (('organizationName', 'GlobalSign nv-sa'),),
               (('commonName', 'GlobalSign RSA OV SSL CA 2018'),)),
 'notAfter': 'Aug  6 05:16:01 2023 GMT',
 'notBefore': 'Jul  5 05:16:02 2022 GMT',
 'serialNumber': '4417CE86EF82EC6921CC6F68',
 'subject': (((('countryName', 'CN'),),
                 (('stateOrProvinceName', 'beijing'),),
                 (('localityName', 'beijing'),),
                 (('organizationalUnitName', 'service operation department'),),
                 (('organizationName',
                   'Beijing Baidu Netcom Science Technology Co., Ltd'),),
                 (('commonName', 'baidu.com'),)),
 'subjectAltName': (('DNS', 'baidu.com'),
                    ('DNS', 'baifubao.com'),
                    ('DNS', 'www.baidu.cn'),
                    ('DNS', 'www.baidu.com.cn'),
                    ('DNS', 'mct.y.nuomi.com'),
                    ('DNS', 'apollo.auto'),
                    ('DNS', 'dwz.cn'),
                    ('DNS', '*.baidu.com'),
                    ('DNS', '*.baifubao.com'),
                    ('DNS', '*.baidustatic.com'),
                    ('DNS', '*.bdstatic.com'),
```



```
'version': 3}
[{'issuer': (((('countryName', 'BE'),),
                (('organizationName', 'GlobalSign nv-sa'),),
                (('organizationalUnitName', 'Root CA'),),
                (('commonName', 'GlobalSign Root CA'))),
  'notAfter': 'Jan 28 12:00:00 2028 GMT',
  'notBefore': 'Sep 1 12:00:00 1998 GMT',
  'serialNumber': '040000000001154B5AC394',
  'subject': (((('countryName', 'BE'),),
                 (('organizationName', 'GlobalSign nv-sa'),),
                 (('organizationalUnitName', 'Root CA'),),
                 (('commonName', 'GlobalSign Root CA'))),
  'version': 3}]
```


3. 请同学们将修改 www.baidu.com 网站主机名的测试过程截图保存在报告里并分析执行的结果，也可选用其他网站做测试。

当 context.check_hostname = True 时，执行 ./handshake.py

www.baidu.com 会有报错信息，显示、Hostname 不匹配，如下图所示：

```
root@743cc08b925f:/volumes# echo 14.119.104.254 www.baidul.com >> /etc/hosts
root@743cc08b925f:/volumes# ./handshake.py www.baidul.com
After making TCP connection. Press any key to continue ...
Traceback (most recent call last):
  File "./handshake.py", line 29, in <module>
    ssock.do_handshake() # Start the handshake
  File "/usr/lib/python3.8/ssl.py", line 1309, in do_handshake
    self._sslobj.do_handshake()
ssl.SSLCertVerificationError: [SSL: CERTIFICATE_VERIFY_FAILED] certificate ver
ify failed: Hostname mismatch, certificate is not valid for 'www.baidul.com'.
(_ssl.c:1123)
```

当 context.check_hostname = False 时，执行 ./handshake.py

www.baidu.com 显示正常，如下图所示：

```
root@743cc08b925f:/volumes# ./handshake.py www.baidu1.com
After making TCP connection. Press any key to continue ...
=== Cipher used: ('ECDHE-RSA-AES128-GCM-SHA256', 'TLSv1.2', 128)
=== Server hostname: www.baidu1.com
=== Server certificate:
{'OCSP': ('http://ocsp.globalsign.com/gsrsoovsslca2018',),
 'caIssuers': ('http://secure.globalsign.com/cacert/gsrsoovsslca2018.crt',),
 'crlDistributionPoints': ('http://crl.globalsign.com/gsrsoovsslca2018.crl',
 'issuer': (((('countryName', 'BE'),),
               (('organizationName', 'GlobalSign nv-sa'),),
               (('commonName', 'GlobalSign RSA OV SSL CA 2018'),)),
 'notAfter': 'Aug  6 05:16:01 2023 GMT',
 'notBefore': 'Jul  5 05:16:02 2022 GMT',
 'serialNumber': '4417CE86EF82EC6921CC6F68',
 'subject': (((('countryName', 'CN'),),
                (('stateOrProvinceName', 'beijing'),),
                (('localityName', 'beijing'),),
                (('organizationalUnitName', 'service operation department'),),
                (('organizationName',
                  'Beijing Baidu Netcom Science Technology Co., Ltd'),),
                (('commonName', 'baidu.com'),)),
 'subjectAltName': (('DNS', 'baidu.com'),
                    ('DNS', 'baifubao.com'),
                    ('DNS', 'www.baidu.cn'),
                    ('DNS', 'www.baidu.com.cn'),
                    ('DNS', 'mct.y.nuomi.com'),
                    ('DNS', 'apollo.auto'),
                    ('DNS', 'dwz.cn'),
                    ('DNS', '*.baidu.com'),
                    ('DNS', '*.baifubao.com'),
                    ('DNS', '*.baidustatic.com'),
```

```

('DNS', '*.bdstatic.com'),
('DNS', '*.bdimg.com'),
('DNS', '*.hao123.com'),
('DNS', '*.nuomi.com'),
('DNS', '*.chuanke.com'),
('DNS', '*.trustgo.com'),
('DNS', '*.bce.baidu.com'),
('DNS', '*.eyun.baidu.com'),
('DNS', '*.map.baidu.com'),
('DNS', '*.mbd.baidu.com'),
('DNS', '*.fanyi.baidu.com'),
('DNS', '*.baidubce.com'),
('DNS', '*.mipcdn.com'),
('DNS', '*.news.baidu.com'),
('DNS', '*.baidupcs.com'),
('DNS', '*.aipage.com'),
('DNS', '*.aipage.cn'),
('DNS', '*.bcehost.com'),
('DNS', '*.safe.baidu.com'),
('DNS', '*.im.baidu.com'),
('DNS', '*.baiducontent.com'),
('DNS', '*.dlnel.com'),
('DNS', '*.dlnel.org'),
('DNS', '*.dueros.baidu.com'),
('DNS', '*.su.baidu.com'),
('DNS', '*.91.com'),
('DNS', '*.hao123.baidu.com'),
('DNS', '*.apollo.auto'),
('DNS', '*.xueshu.baidu.com'),
('DNS', '*.bj.baidubce.com'),
('DNS', '*.gz.baidubce.com'),
('DNS', '*.smartapps.cn'),

('DNS', '*.smartapps.cn'),
('DNS', '*.bdtjrcv.com'),
('DNS', '*.hao222.com'),
('DNS', '*.haokan.com'),
('DNS', '*.pae.baidu.com'),
('DNS', '*.vd.bdstatic.com'),
('DNS', '*.cloud.baidu.com'),
('DNS', 'click.hm.baidu.com'),
('DNS', 'log.hm.baidu.com'),
('DNS', 'cm.pos.baidu.com'),
('DNS', 'wn.pos.baidu.com'),
('DNS', 'update.pan.baidu.com')),

'version': 3}
[{'issuer': (((('countryName', 'BE'),),
                (('organizationName', 'GlobalSign nv-sa'),),
                (('organizationalUnitName', 'Root CA'),),
                (('commonName', 'GlobalSign Root CA'),))),
  'notAfter': 'Jan 28 12:00:00 2028 GMT',
  'notBefore': 'Sep 1 12:00:00 1998 GMT',
  'serialNumber': '040000000001154B5AC394',
  'subject': (((('countryName', 'BE'),),
                  (('organizationName', 'GlobalSign nv-sa'),),
                  (('organizationalUnitName', 'Root CA'),),
                  (('commonName', 'GlobalSign Root CA'),))),
  'version': 3}]
After TLS handshake. Press any key to continue ...
root@743cc08b925f:/volumes# █

```

这是因为，如果 `context.check_hostname = False`，当服务器返回"Server Hello"消息时，客户端将不会对服务器的证书中的 `hostname` 进行校验。在此示例中，服务器证书主机名应该是 `www.baidu.com`，但由于客户端不进行主机名校验，所以可以接受证书中的主机名为 `www.baidu1.com`，并与服务器建立连接。

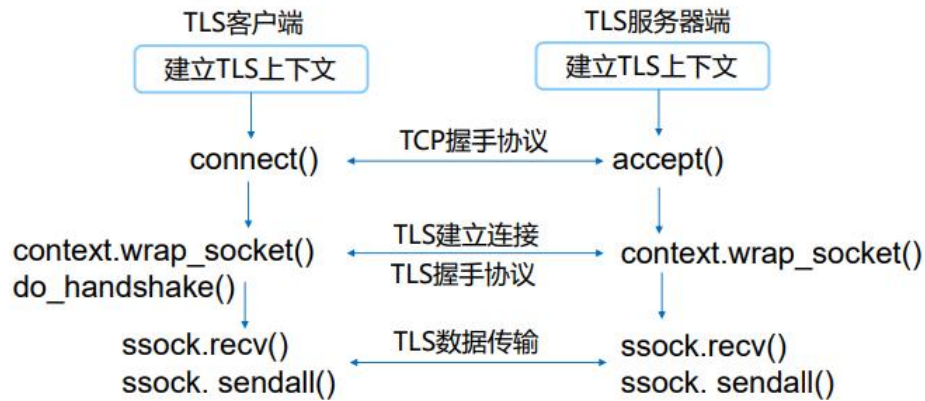
4. 请分析 TLS 客户端编程和 `server.py` 的代码，说明客户端和服务端程序的关键步骤。

利用 `handshake.p` 修改 `client.py`，并增加相应代码：

```
29 ssock.do_handshake()    # Start the handshake
30 print("=== Cipher used: {}".format(ssock.cipher()))
31 print("=== Server hostname: {}".format(ssock.server_hostname))
32 print("=== Server certificate:")
33 pprint.pprint(ssock.getpeercert())
34 pprint.pprint(context.get_ca_certs())
35 #input("After TLS handshake. Press any key to continue ...")
36
37 # Send HTTP Request to Server
38 request = b"GET / HTTP/1.0\r\nHost: " + hostname.encode('utf-8') +
39          b"\r\n\r\n"
40 # Read HTTP Response from Server
41 response = ssock.recv(2048)
42 while response:
43     pprint.pprint(response.split(b"\r\n"))
44     response = ssock.recv(2048)
45
46 # Close the TLS Connection
47 ssock.shutdown(socket.SHUT_RDWR)
48 ssock.close()
49
50
```

分析代码可知，客户端和服务端程序的关键步骤为，双方首先利用 `connect()` 函数和 `accept()` 函数确定 TCP 握手协议，然后利用 `context.wrap_socket()` 函数和 `do_handshake()` 函数进行 TLS 建立连接与 TLS 握手协议，最后利用 `ssock.recv()` 函数和 `ssock.sendall()` 函数进行 TLS 数据传输。

过程如下图所示：



5.请分别用 client.py 和浏览器两种方式访问服务器，并记录你观察的结果（截图）

用 client.py 访问服务器的结果如下：

客户端处的结果：

```
root@bf220f70d4d6:/volumes# ./server.py
Enter PEM pass phrase:
TLS connection established
"Request: b'GET / HTTP/1.0\\r\\nHost: www.bank32.com\\r\\n\\r\\n'"
```

服务器处的结果：

```

root@743cc08b925f:/volumes# ./client.py www.bank32.com
After making TCP connection. Press any key to continue ...
=== Cipher used: ('TLS_AES_256_GCM_SHA384', 'TLSv1.3', 256)
=== Server hostname: www.bank32.com
=== Server certificate:
{'issuer': (((('commonName', 'www.modelCA.com')),
              (('organizationName', 'Model CA LTD.'))),
              (('countryName', 'US'))),
 'notAfter': 'May 22 09:44:52 2033 GMT',
 'notBefore': 'May 25 09:44:52 2023 GMT',
 'serialNumber': '1000',
 'subject': (((('countryName', 'US')),
                (('organizationName', 'Bank32 Inc.'))),
                (('commonName', 'www.bank32.com'))),
 'subjectAltName': (('DNS', 'www.bank32.com'),
                    ('DNS', 'www.bank32A.com'),
                    ('DNS', 'www.bank32B.com')),
 'version': 3}
[{'issuer': (((('commonName', 'www.modelCA.com')),
              (('organizationName', 'Model CA LTD.'))),
              (('countryName', 'US'))),
 'notAfter': 'May 22 09:43:48 2033 GMT',
 'notBefore': 'May 25 09:43:48 2023 GMT',
 'serialNumber': '1EC59475CA1D0161F30983054325D2E1CB601EDD',
 'subject': (((('commonName', 'www.modelCA.com')),
                (('organizationName', 'Model CA LTD.'))),
                (('countryName', 'US'))),
 'version': 3}]
[b'\nHTTP/1.1 200 OK',

[b'\nHTTP/1.1 200 OK',
 b'Content-Type: text/html',
 b'',
 b'\n<!DOCTYPE html><html><body><h1>This is Bank32.com!</h1></body></html>\n']
root@743cc08b925f:/volumes# █

```

用浏览器访问服务器的结果如下：



