项目编号： 22

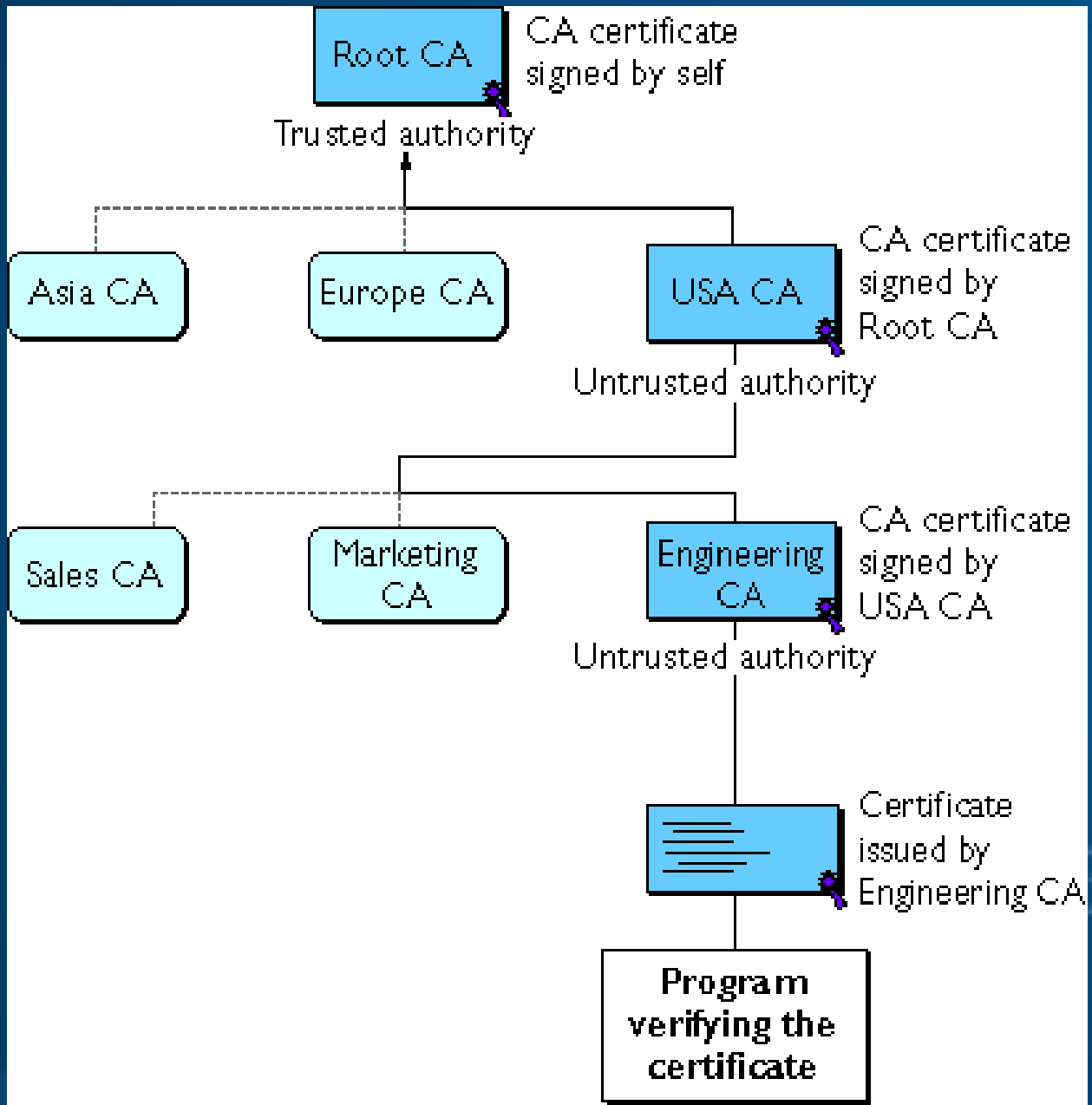
利用OpenSSL实现证书的创建和验证

组长：吴道远

组员：邵泽慧　徐林　邵依　陈浩

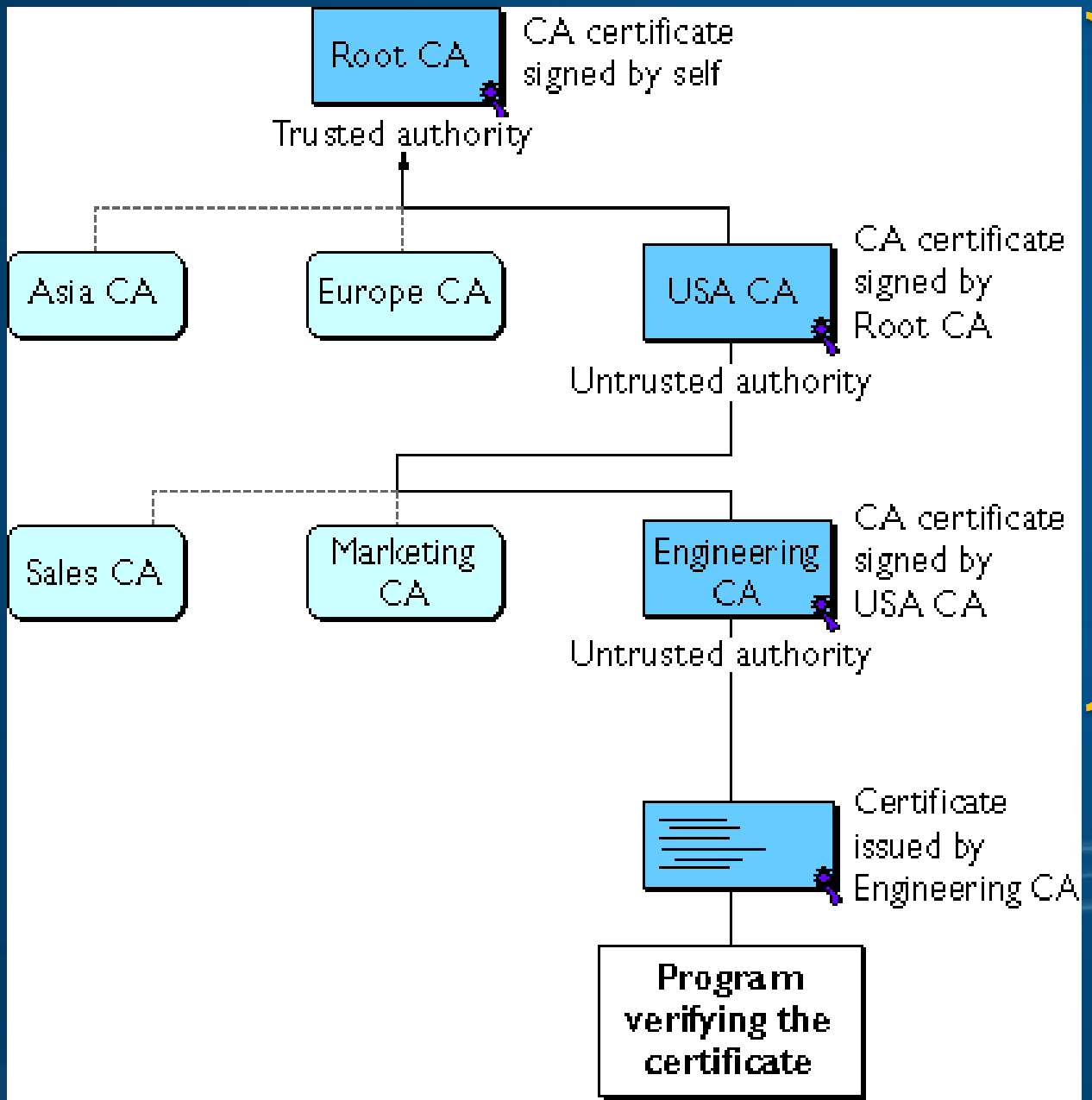
一、创建证书

证书的层次结构



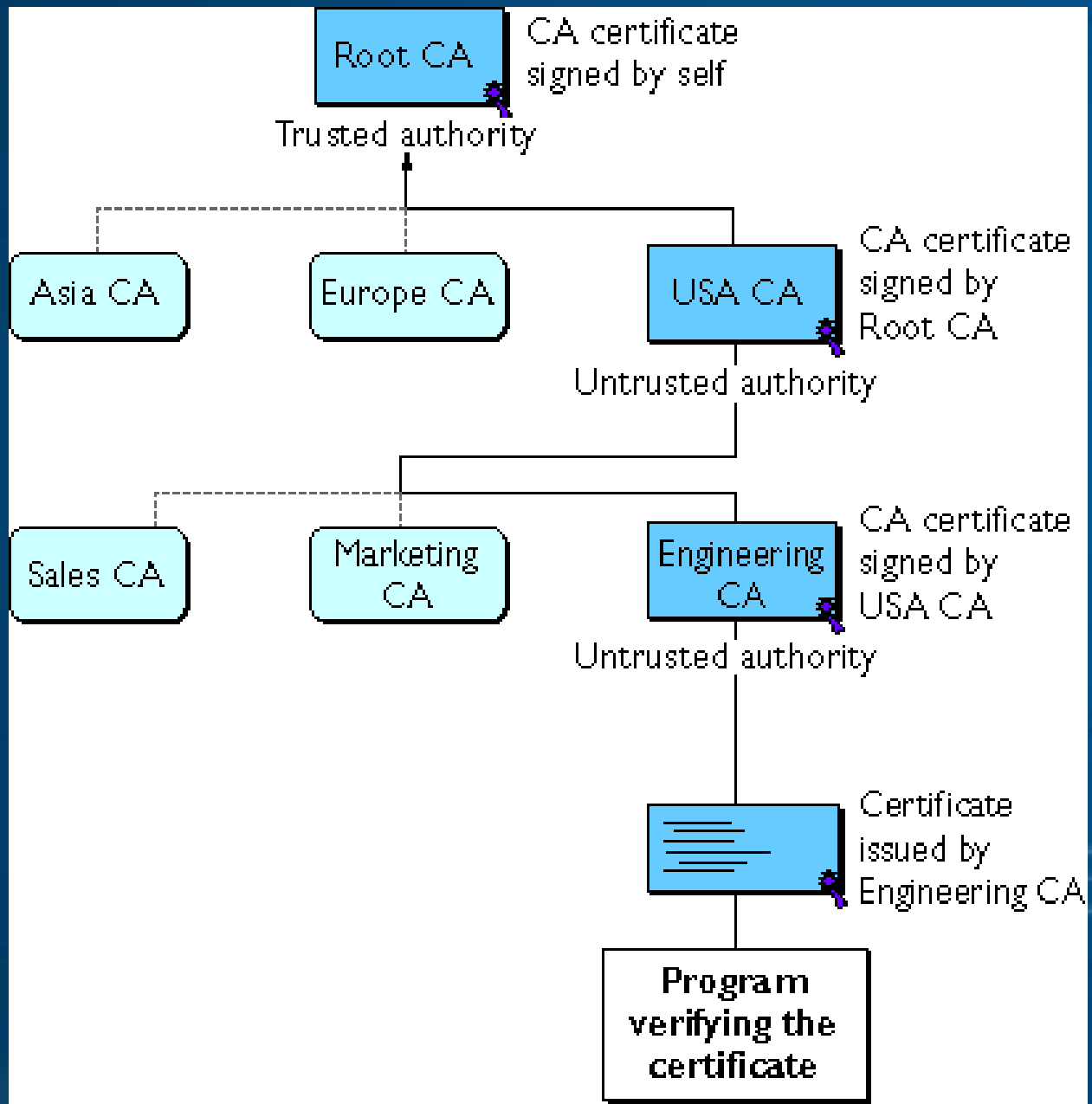
一、创建证书

证书的层次结构



一、创建证书

证书的层次结构



一、创建证书

创建根证书(Root CA，最高层认证中心)

1. 创建CA的私钥：

```
openssl.exe genrsa -out root\root-key.pem 1024
```

2. 创建CA的证书请求：(clzqwdy@gmail.com)

```
openssl req -new -key root\root-key.pem -out root\root-req.csr -text
```

3. 对刚刚生成的证书请求进行自签名：

```
openssl.exe x509 -req -in root\root-req.csr -out root\root-cert.pem -sha1 -signkey
root\root-key.pem -days 3650 -text -extfile openssl.cnf -extensions v3_ca
```

4. 将证书导出成浏览器支持的.p12格式：(PKCS标准)

```
openssl.exe pkcs12 -export -cacerts -in root\root-cert.pem -inkey root\root-
key.pem -out root\root.p12
```

一、创建证书

创建根证书(Root CA，最高层认证中心)

1. 创建CA的私钥：

```
openssl.exe genrsa -out root\root-key.pem 1024
```

2. 创建CA的证书请求：(clzqwdy@gmail.com)

```
openssl req -new -key root\root-key.pem -out root\root-req.csr -text
```

3. 对刚刚生成的证书请求进行自签名：

```
openssl.exe x509 -req -in root\root-req.csr -out root\root-cert.pem -sha1 -signkey
root\root-key.pem -days 3650 -text -extfile openssl.cnf -extensions v3_ca
```

4. 将证书导出成浏览器支持的.p12格式：(PKCS标准)

```
openssl.exe pkcs12 -export -cacerts -in root\root-cert.pem -inkey root\root-
key.pem -out root\root.p12
```

一、创建证书

创建根证书(Root CA，最高层认证中心)

1. 创建CA的私钥：

```
openssl.exe genrsa -out root\root-key.pem 1024
```

2. **创建CA的证书请求：**(clzqwdy@gmail.com)

```
openssl req -new -key root\root-key.pem -out root\root-req.csr -text
```

3. 对刚刚生成的证书请求进行**自签名**：

```
openssl.exe x509 -req -in root\root-req.csr -out root\root-cert.pem -sha1 -signkey
root\root-key.pem -days 3650 -text -extfile openssl.cnf -extensions v3_ca
```

4. 将证书导出成浏览器支持的.p12格式：(PKCS标准)

```
openssl.exe pkcs12 -export -cacerts -in root\root-cert.pem -inkey root\root-
key.pem -out root\root.p12
```

一、创建证书

创建根证书(Root CA，最高层认证中心)

1. 创建CA的私钥：

```
openssl.exe genrsa -out root\root-key.pem 1024
```

2. 创建CA的证书请求：(clzqwdy@gmail.com)

```
openssl req -new -key root\root-key.pem -out root\root-req.csr -text
```

3. 对刚刚生成的证书请求进行自签名：

```
openssl.exe x509 -req -in root\root-req.csr -out root\root-cert.pem -sha1 -signkey
root\root-key.pem -days 3650 -text -extfile openssl.cnf -extensions v3_ca
```

4. 将证书导出成浏览器支持的.p12格式：(PKCS标准)

```
openssl.exe pkcs12 -export -cacerts -in root\root-cert.pem -inkey root\root-
key.pem -out root\root.p12
```

一、创建证书

创建根证书(Root CA，最高层认证中心)

1. 创建CA的私钥：

```
openssl.exe genrsa -out root\root-key.pem 1024
```

2. 创建CA的证书请求：(clzqwdy@gmail.com)

```
openssl req -new -key root\root-key.pem -out root\root-req.csr -text
```

3. 对刚刚生成的证书请求进行自签名：

```
openssl.exe x509 -req -in root\root-req.csr -out root\root-cert.pem -sha1 -signkey
root\root-key.pem -days 3650 -text -extfile openssl.cnf -extensions v3_ca
```

4. 将证书导出成浏览器支持的.p12格式：(PKCS标准)

```
openssl.exe pkcs12 -export -cacerts -in root\root-cert.pem -inkey root\root-
key.pem -out root\root.p12
```

一、创建证书

创建Server证书

1. 创建Server的私钥：

```
openssl.exe genrsa -out server\server-key.pem 1024
```

2. 创建Server的证书请求：(hacker@gmail.com)

```
openssl req -new -key server\server-key.pem -out server\server-req.csr -text
```

3. 用CA的证书用私钥对Server的证书请求进行签名：

```
openssl.exe x509 -req -in server\server-req.csr -CA root\root-cert.pem -CAkey
root\root-key.pem -CAcreateserial -days 730 -out server\server-cert.pem -text
```

4. 将证书导出成浏览器支持的.p12格式：

```
openssl.exe pkcs12 -export -clcerts -in root\root-cert.pem -inkey root\root-
key.pem -out root\root.p12
```

一、创建证书

创建Server证书

1. 创建Server的私钥：

```
openssl.exe genrsa -out server\server-key.pem 1024
```

2. 创建Server的证书请求：(hacker@gmail.com)

```
openssl req -new -key server\server-key.pem -out server\server-req.csr -text
```

3. 用CA的证书用私钥对Server的证书请求进行签名：

```
openssl.exe x509 -req -in server\server-req.csr -CA root\root-cert.pem -CAkey
root\root-key.pem -CAcreateserial -days 730 -out server\server-cert.pem -text
```

4. 将证书导出成浏览器支持的.p12格式：

```
openssl.exe pkcs12 -export -clcerts -in root\root-cert.pem -inkey root\root-
key.pem -out root\root.p12
```

一、创建证书

创建Server证书

1. 创建Server的私钥：

```
openssl.exe genrsa -out server\server-key.pem 1024
```

2. **创建Server的证书请求：** (hacker@gmail.com)

```
openssl req -new -key server\server-key.pem -out server\server-req.csr -text
```

3. 用CA的证书用私钥对Server的证书请求进行签名：

```
openssl.exe x509 -req -in server\server-req.csr -CA root\root-cert.pem -CAkey
root\root-key.pem -CAcreateserial -days 730 -out server\server-cert.pem -text
```

4. 将证书导出成浏览器支持的.p12格式：

```
openssl.exe pkcs12 -export -clcerts -in root\root-cert.pem -inkey root\root-
key.pem -out root\root.p12
```

一、创建证书

创建Server证书

1. 创建Server的私钥：

```
openssl.exe genrsa -out server\server-key.pem 1024
```

2. 创建Server的证书请求：(hacker@gmail.com)

```
openssl req -new -key server\server-key.pem -out server\server-req.csr -text
```

3. 用CA的证书用私钥对Server的证书请求进行签名：

```
openssl.exe x509 -req -in server\server-req.csr -CA root\root-cert.pem -CAkey
root\root-key.pem -CAcreateserial -days 730 -out server\server-cert.pem -text
```

4. 将证书导出成浏览器支持的.p12格式：

```
openssl.exe pkcs12 -export -clcerts -in root\root-cert.pem -inkey root\root-
key.pem -out root\root.p12
```

一、创建证书

创建Server证书

1. 创建Server的私钥：

```
openssl.exe genrsa -out server\server-key.pem 1024
```

2. 创建Server的证书请求：(hacker@gmail.com)

```
openssl req -new -key server\server-key.pem -out server\server-req.csr -text
```

3. 用CA的证书用私钥对Server的证书请求进行签名：

```
openssl.exe x509 -req -in server\server-req.csr -CA root\root-cert.pem -CAkey
root\root-key.pem -CAcreateserial -days 730 -out server\server-cert.pem -text
```

4. **将证书导出成浏览器支持的.p12格式：**

```
openssl.exe pkcs12 -export -clcerts -in root\root-cert.pem -inkey root\root-
key.pem -out root\root.p12
```

二、证书格式

1. X.509 标准

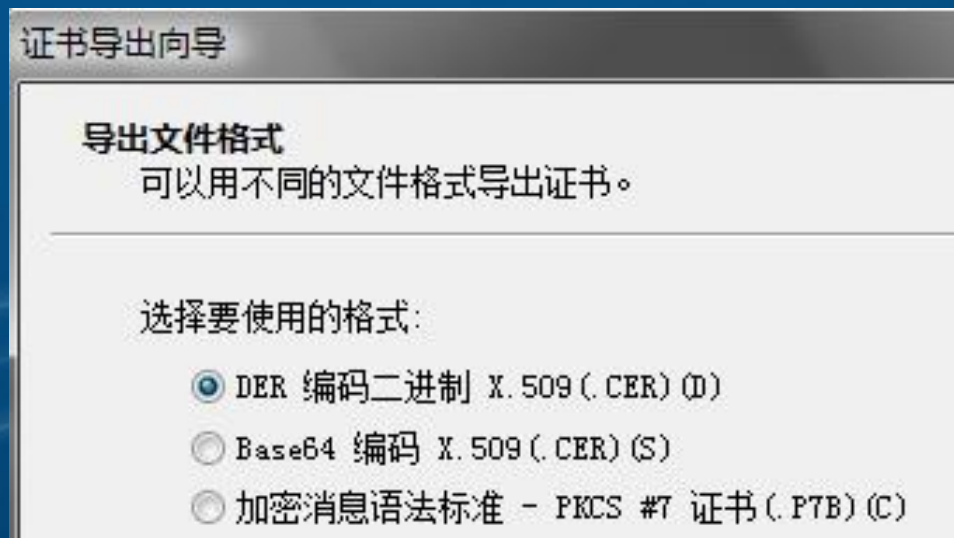
a) 类比于POSIX，反正就是个接口标准
b) 大多数证书格式遵循的. Also: X.500 PKCS
c) 结构!!! 见server-cert.pem

2. .pem后缀的证书

a) PEM格式(-inform PEM)，Base64编码
b) -----BEGIN CERTIFICATE----- 与 -----END CERTIFICATE-----

3. .cer .crt .der后缀的证书

a) Usually: DER form，binary
b) Sometimes: Base64 (or PEM form)



二、证书格式

1. X.509 标准

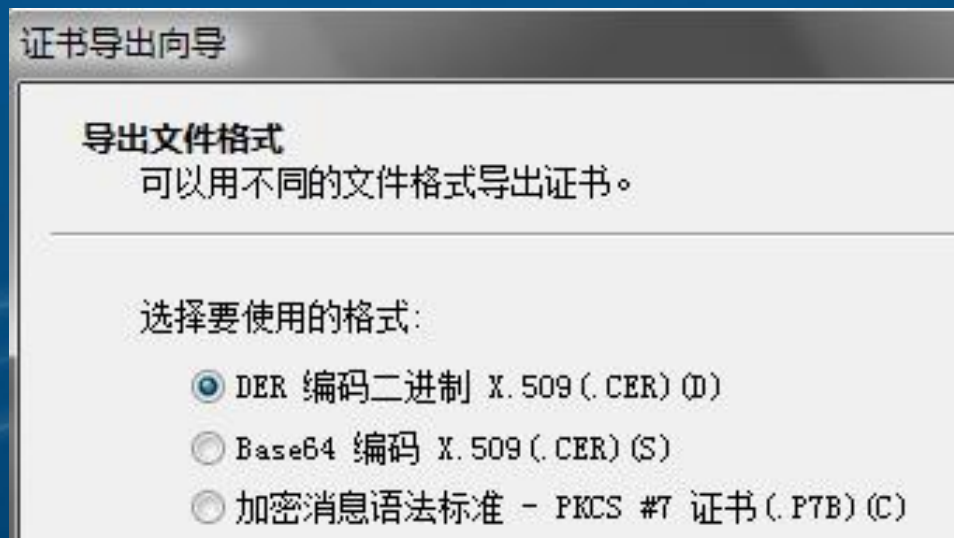
a) 类比于POSIX，反正就是个接口标准
b) 大多数证书格式遵循的. Also: X.500 PKCS
c) 结构!!! 见server-cert.pem

2. .pem后缀的证书

a) PEM格式(-inform PEM)，Base64编码
b) -----BEGIN CERTIFICATE----- 与 -----END CERTIFICATE-----

3. .cer .crt .der后缀的证书

a) Usually: DER form，binary
b) Sometimes: Base64 (or PEM form)



二、证书格式

1. X.509 标准

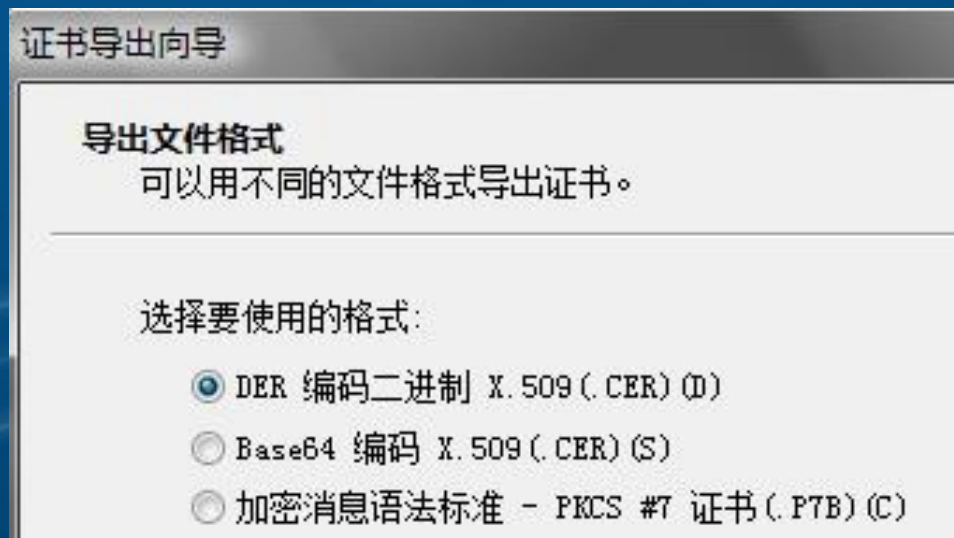
a) 类比于POSIX，反正就是个接口标准
b) 大多数证书格式遵循的. Also: X.500 PKCS
c) 结构!!! 见server-cert.pem

2. .pem后缀的证书

a) PEM格式(-inform PEM)，Base64编码
b) -----BEGIN CERTIFICATE----- 与 -----END CERTIFICATE-----

3. .cer .crt .der后缀的证书

a) Usually: DER form，binary
b) Sometimes: Base64 (or PEM form)



二、证书格式

1. X.509 标准

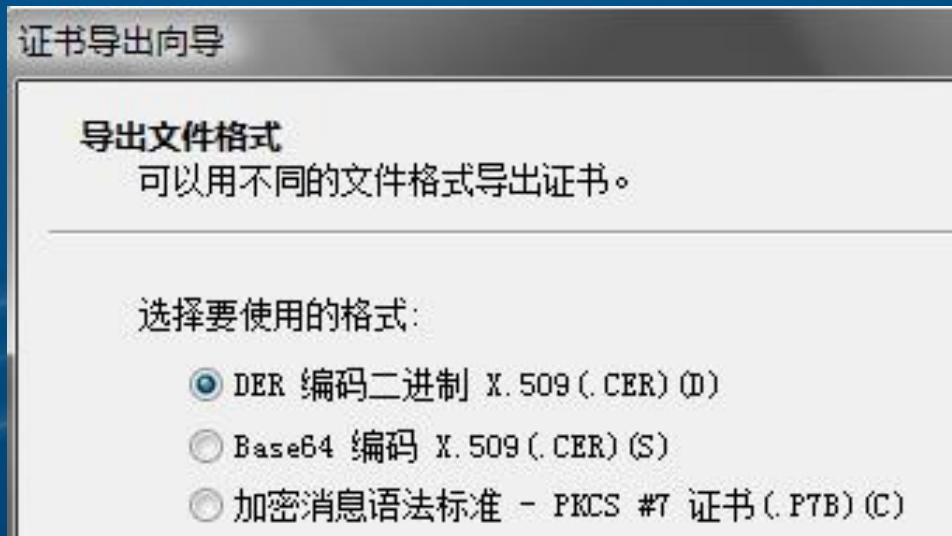
a) 类比于POSIX，反正就是个接口标准
b) 大多数证书格式遵循的. Also: X.500 PKCS
c) 结构!!! 见server-cert.pem

2. .pem后缀的证书

a) PEM格式(-inform PEM)，Base64编码
b) -----BEGIN CERTIFICATE----- 与 -----END CERTIFICATE-----

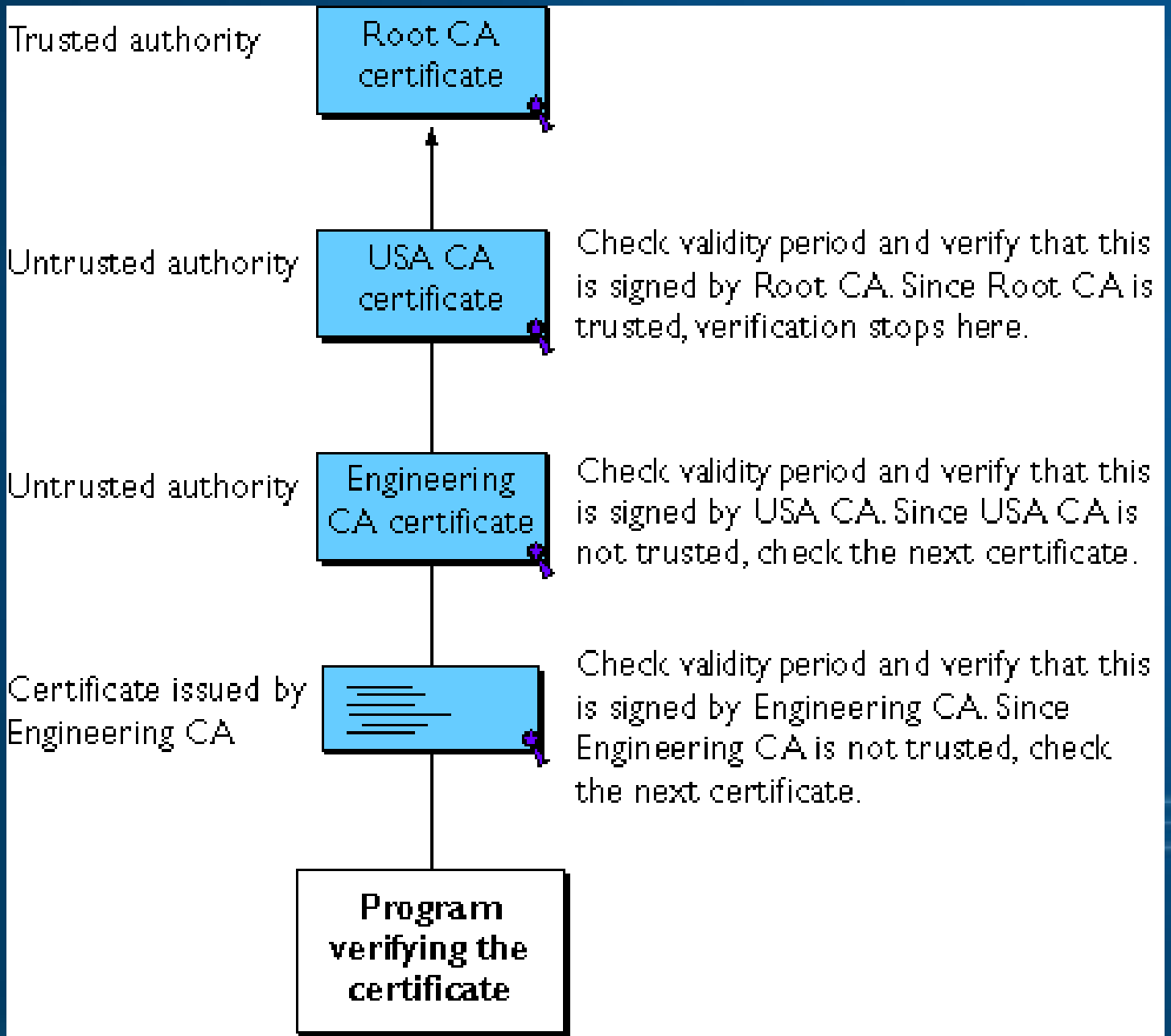
3. .cer .crt .der后缀的证书

a) Usually: DER form，binary
b) Sometimes: Base64 (or PEM form)



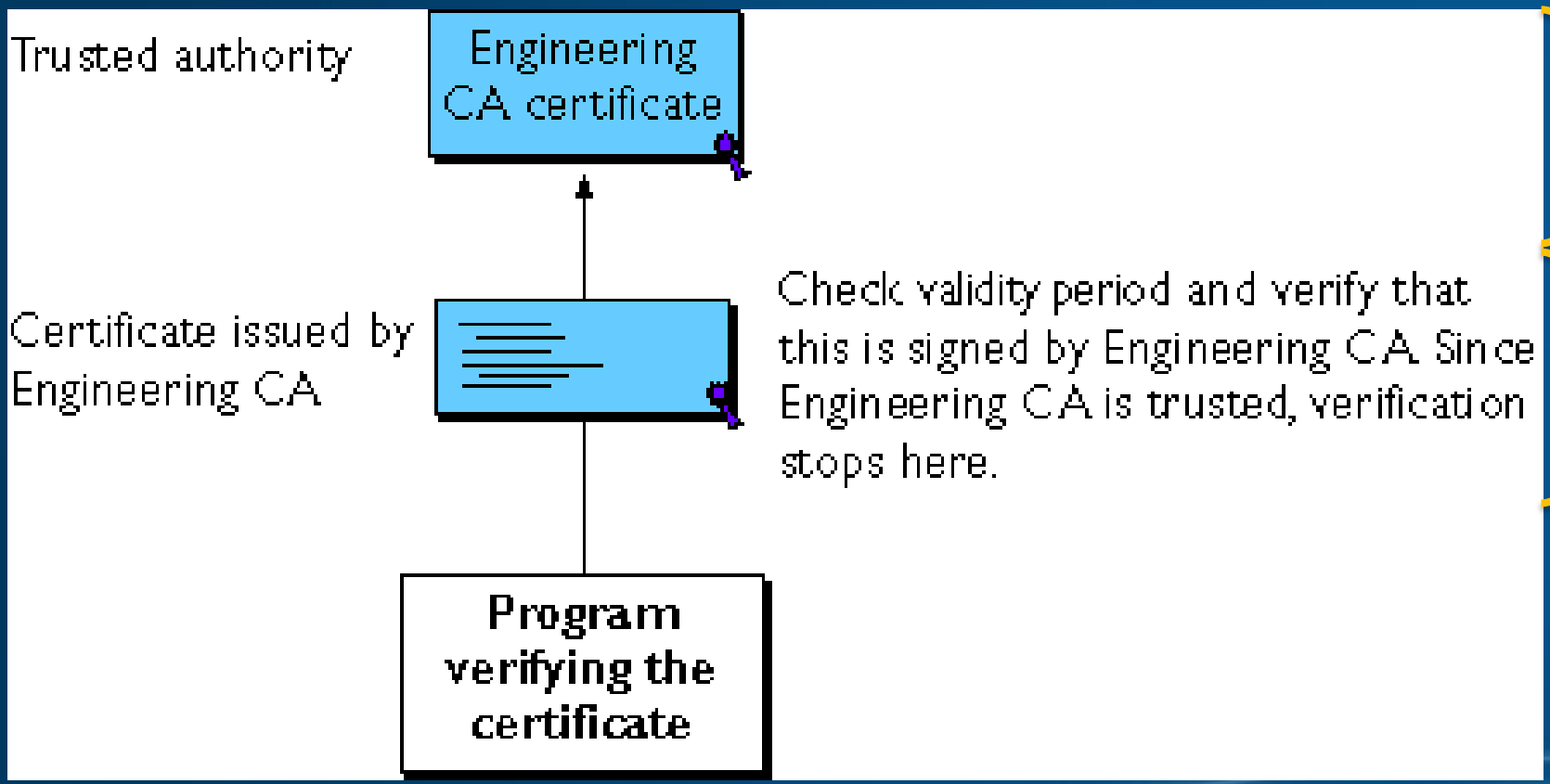
三、验证证书

Verifying a certificate chain all the way to the root CA



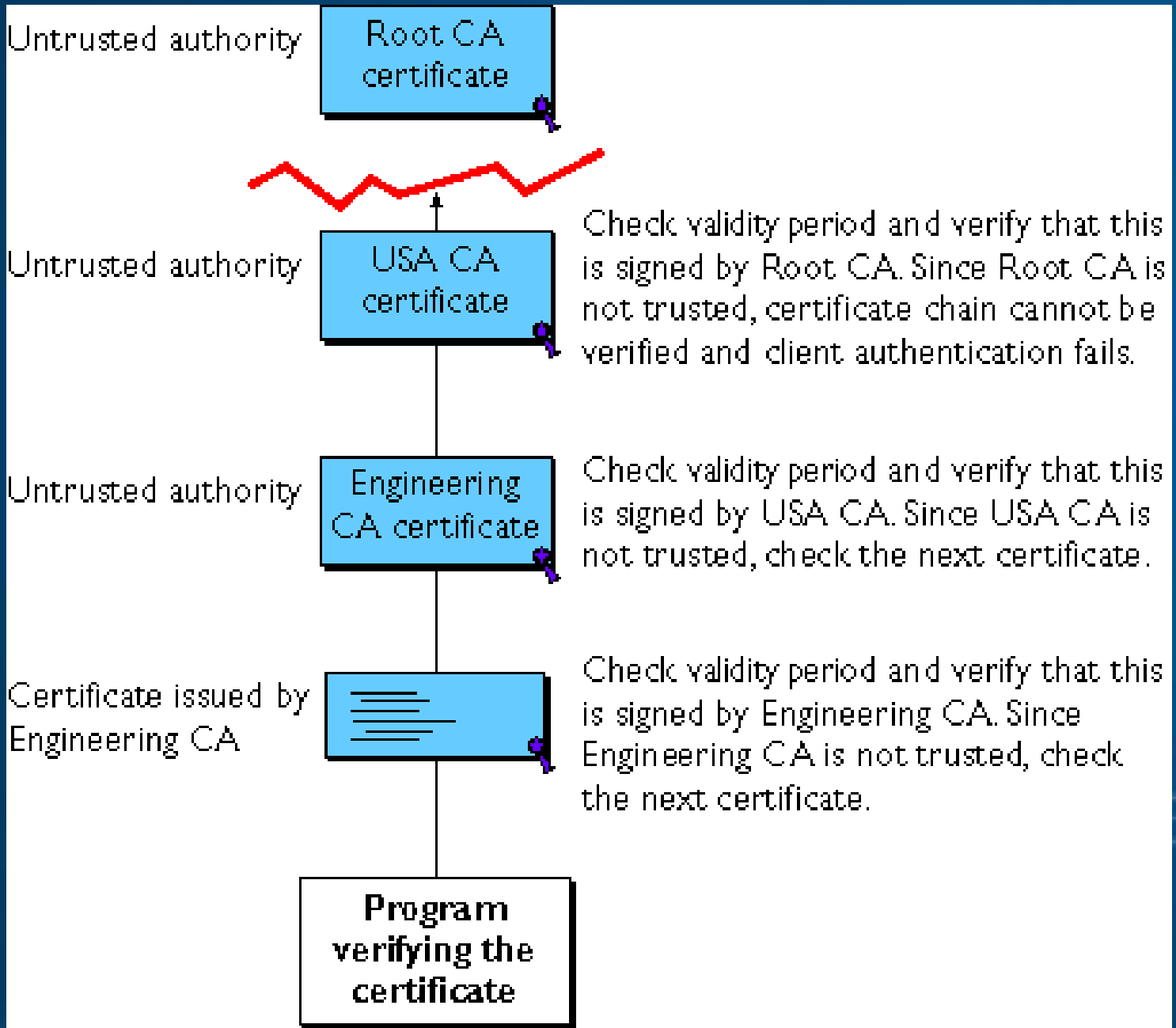
三、验证证书

Verifying a certificate chain to an intermediate CA



三、验证证书

A certificate chain that can't be verified



三、验证证书

demo原理 + 命令行方式验证 + 编程方式验证

1. 结合课上讲的原理demo一下：

a) root-cert.pem AND server-cert.pem
b) 是否过期 + 签名是否正确(即是否是这个CA签的) + CA是否可信
c) Demo: 验证签名是否正确的具体过程

2. 用CA的证书来验证Server的证书：

```
openssl verify -CAfile root\root-cert.pem server\server-cert.pem
```

3. 验证一个已经过期的证书：(and编程验证)

```
openssl verify -CAfile iTrustchina-root-base64.cer iTrustchina-server-base64.cer
```

```
C:\Users\clzqwdy\我的信息\课堂资料\大三下\网络安全\myCert>openssl verify -CAfile
 iTrustchina-root-base64.cer iTrustchina-server-base64.cer
iTrustchina-server-base64.cer: O = "iTruschina Co., Ltd.", OU = Chinese Trust Ne
twork, CN = iTruschina CN Enterprise CA-2
error 10 at 0 depth lookup:certificate has expired
OK
```

三、验证证书

demo原理 + 命令行方式验证 + 编程方式验证

1. 结合课上讲的原理demo一下：

a) root-cert.pem AND server-cert.pem
b) 是否过期 + 签名是否正确(即是否是这个CA签的) + CA是否可信
c) Demo: 验证签名是否正确的具体过程

2. 用CA的证书来验证Server的证书：

```
openssl verify -CAfile root\root-cert.pem server\server-cert.pem
```

3. 验证一个已经过期的证书：(and编程验证)

```
openssl verify -CAfile iTrustchina-root-base64.cer iTrustchina-server-base64.cer
```

```
C:\Users\clzqwdy\我的信息\课堂资料\大三下\网络安全\myCert>openssl verify -CAfile
 iTrustchina-root-base64.cer iTrustchina-server-base64.cer
iTrustchina-server-base64.cer: O = "iTruschina Co., Ltd.", OU = Chinese Trust Ne
twork, CN = iTruschina CN Enterprise CA-2
error 10 at 0 depth lookup:certificate has expired
OK
```

三、验证证书

demo原理 + 命令行方式验证 + 编程方式验证

1. 结合课上讲的原理demo一下：

a) root-cert.pem AND server-cert.pem
b) 是否过期 + 签名是否正确(即是否是这个CA签的) + CA是否可信
c) Demo: 验证签名是否正确的具体过程

2. 用CA的证书来验证Server的证书：

```
openssl verify -CAfile root\root-cert.pem server\server-cert.pem
```

3. 验证一个已经过期的证书：(and编程验证)

```
openssl verify -CAfile iTrustchina-root-base64.cer iTrustchina-server-base64.cer
```

```
C:\Users\clzqwdy\我的信息\课堂资料\大三下\网络安全\myCert>openssl verify -CAfile
 iTrustchina-root-base64.cer iTrustchina-server-base64.cer
iTrustchina-server-base64.cer: O = "iTruschina Co., Ltd.", OU = Chinese Trust Ne
twork, CN = iTruschina CN Enterprise CA-2
error 10 at 0 depth lookup:certificate has expired
OK
```

三、验证证书

demo原理 + 命令行方式验证 + 编程方式验证

1. 结合课上讲的原理demo一下：

a) root-cert.pem AND server-cert.pem
b) 是否过期 + 签名是否正确(即是否是这个CA签的) + CA是否可信
c) Demo: 验证签名是否正确的具体过程

2. 用CA的证书来验证Server的证书：

```
openssl verify -CAfile root\root-cert.pem server\server-cert.pem
```

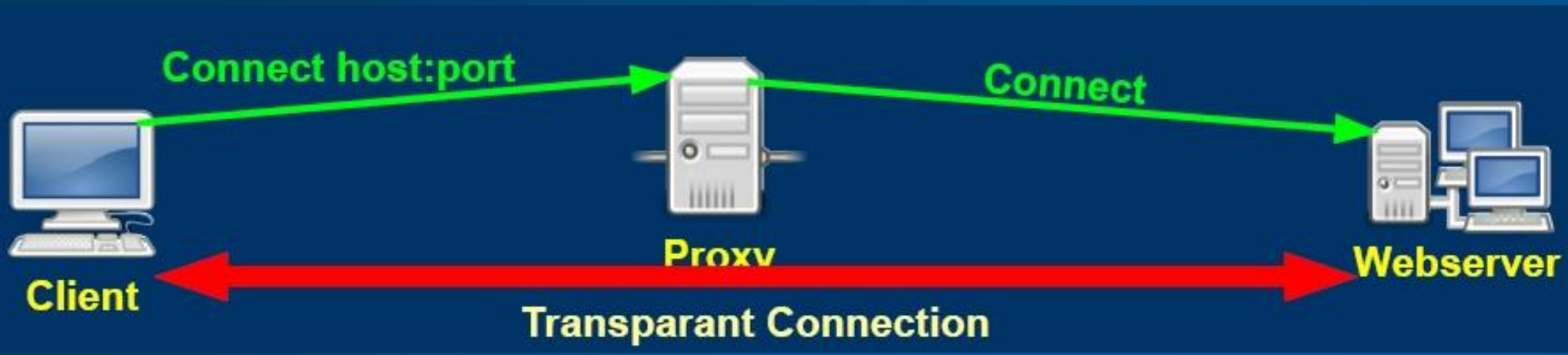
3. 验证一个已经过期的证书：(and编程验证)

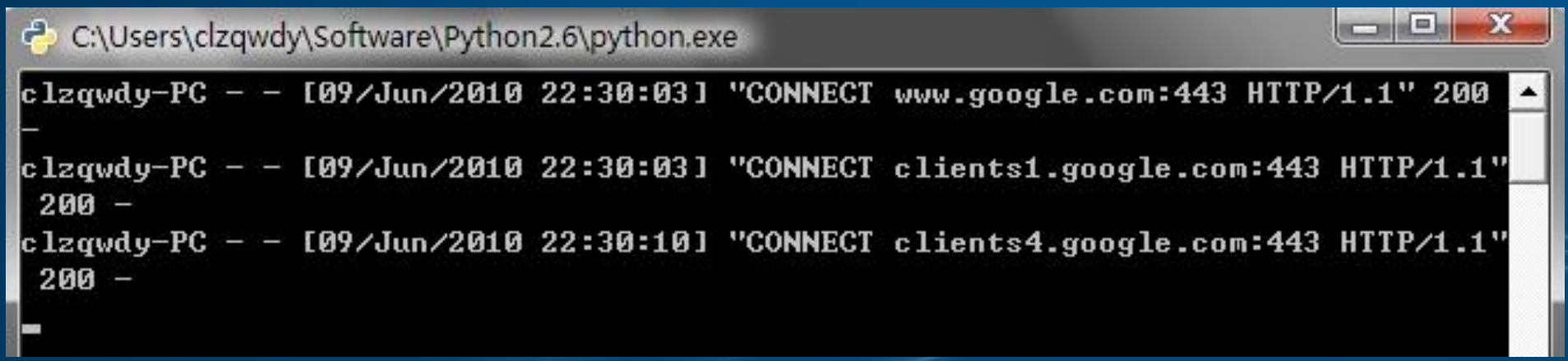
```
openssl verify -CAfile iTrustchina-root-base64.cer iTrustchina-server-base64.cer
```

```
C:\Users\clzqwdy\我的信息\课堂资料\大三下\网络安全\myCert>openssl verify -CAfile
 iTrustchina-root-base64.cer iTrustchina-server-base64.cer
iTrustchina-server-base64.cer: O = "iTruschina Co., Ltd.", OU = Chinese Trust Ne
twork, CN = iTruschina CN Enterprise CA-2
error 10 at 0 depth lookup:certificate has expired
OK
```

四、Http代理

一个正常的Http Proxy处理CONNECT()时的情况





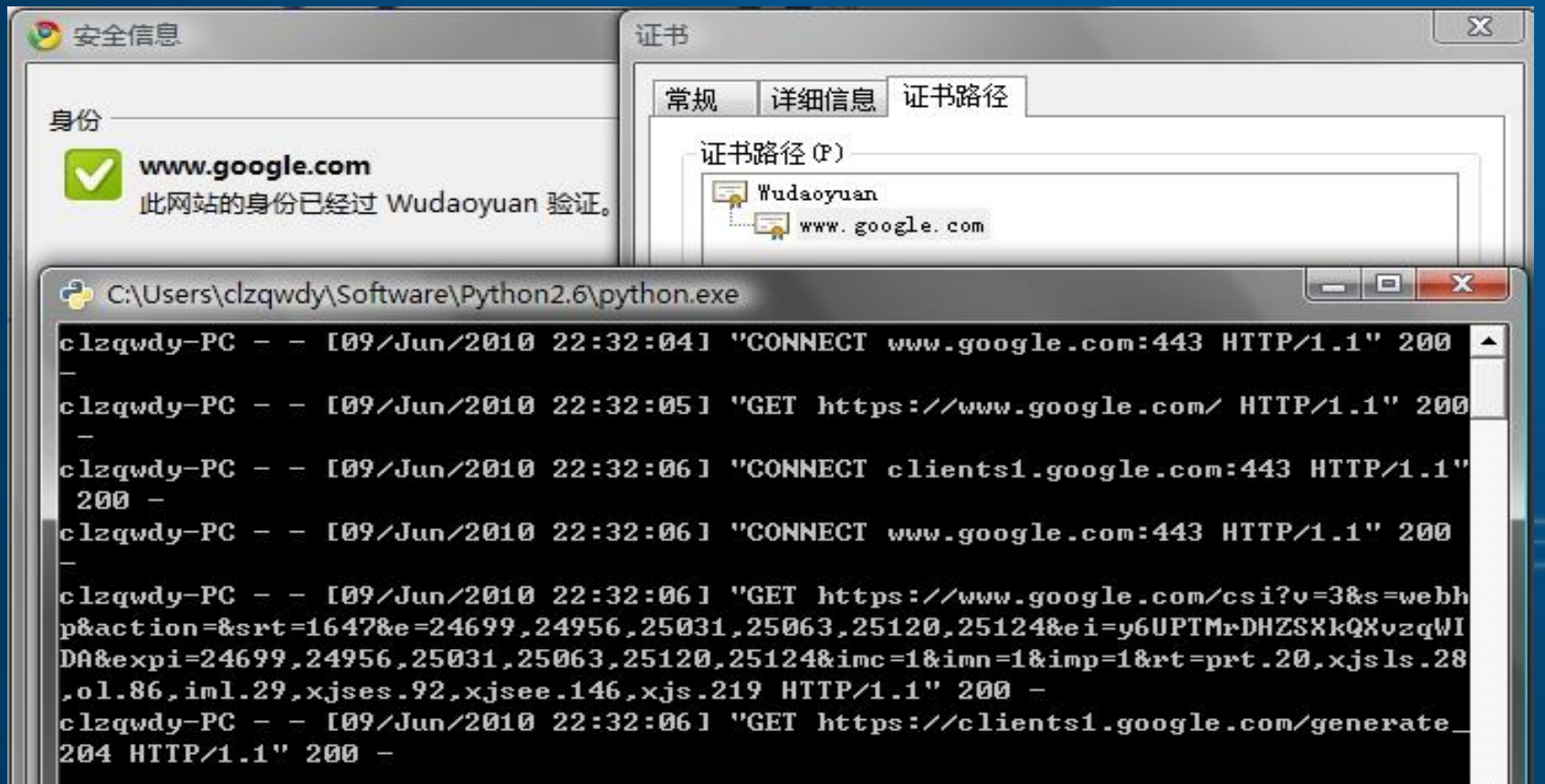
四、Http代理

Man In the Middle

Client

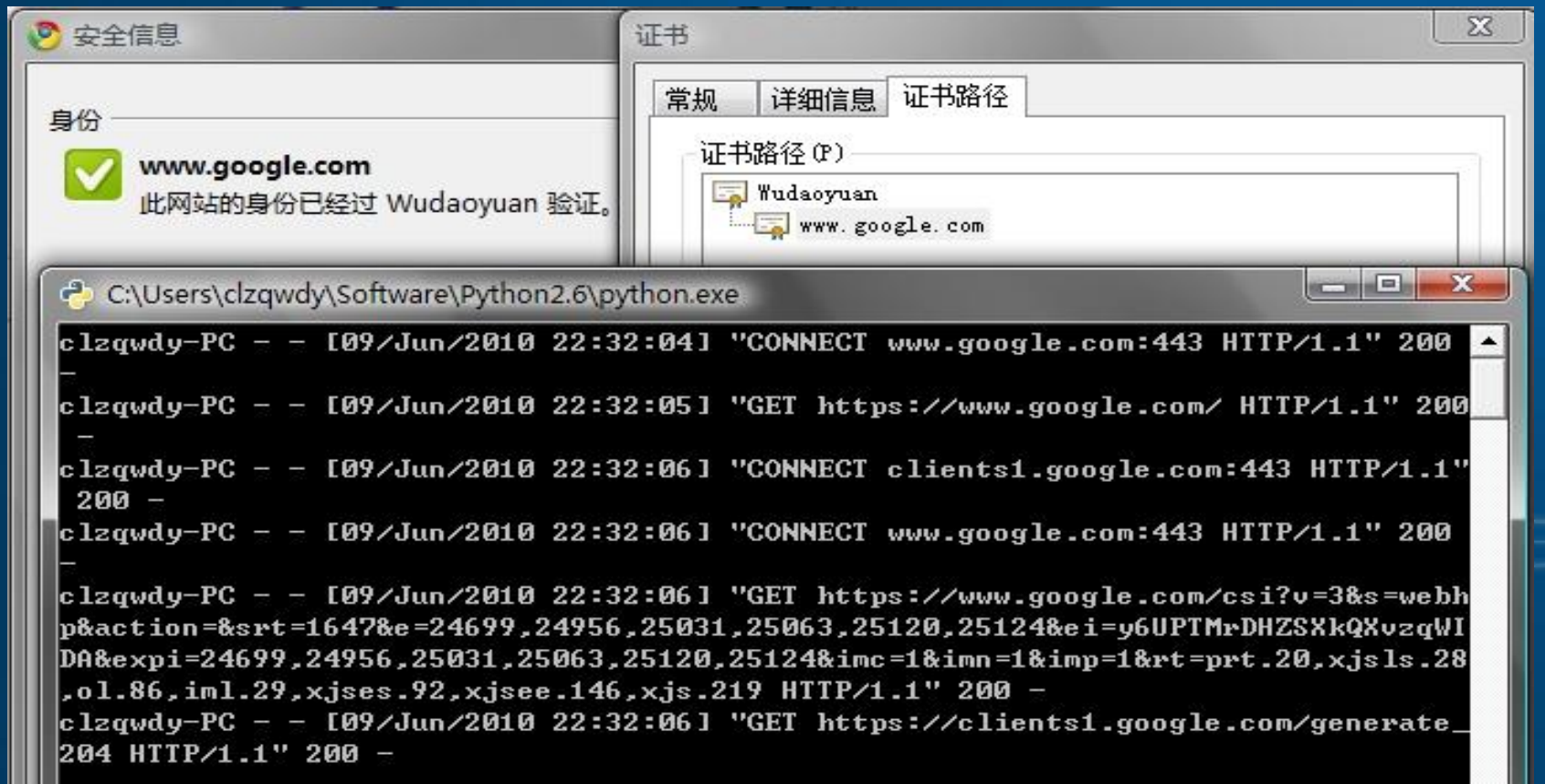
Proxy

Web Server



四、Http代理

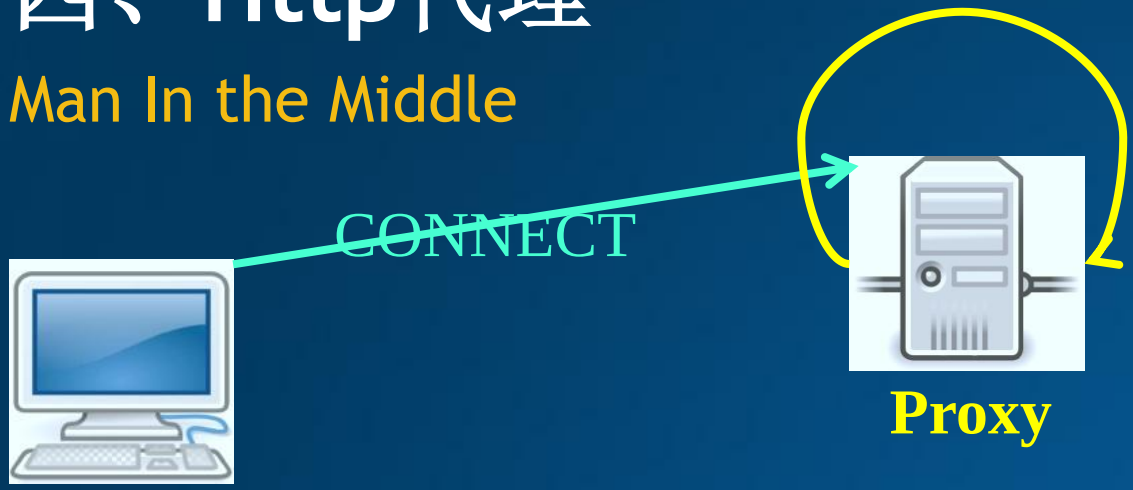
Man In the Middle

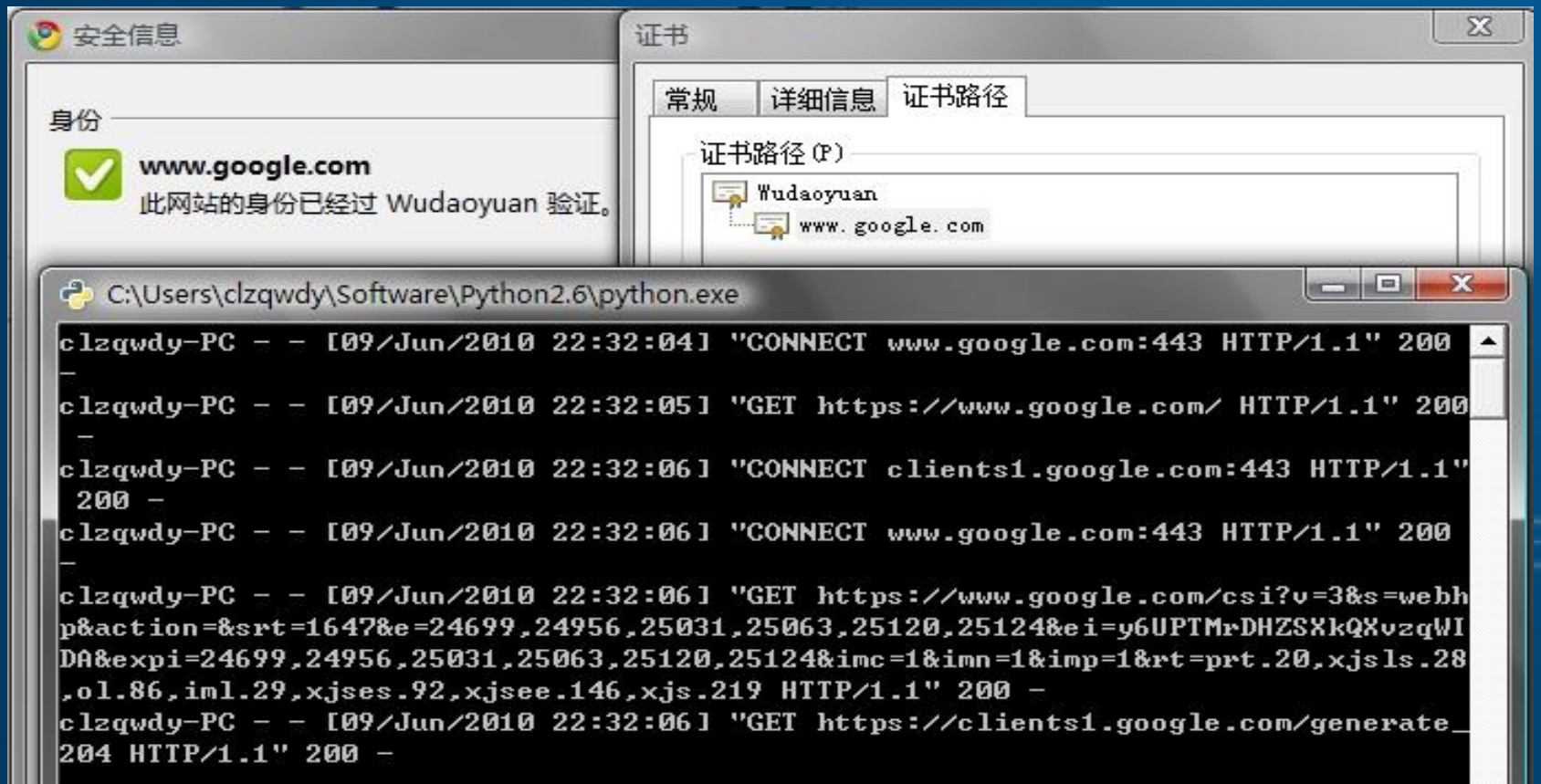


四、Http代理

CA sign a server cert to Host

Man In the Middle

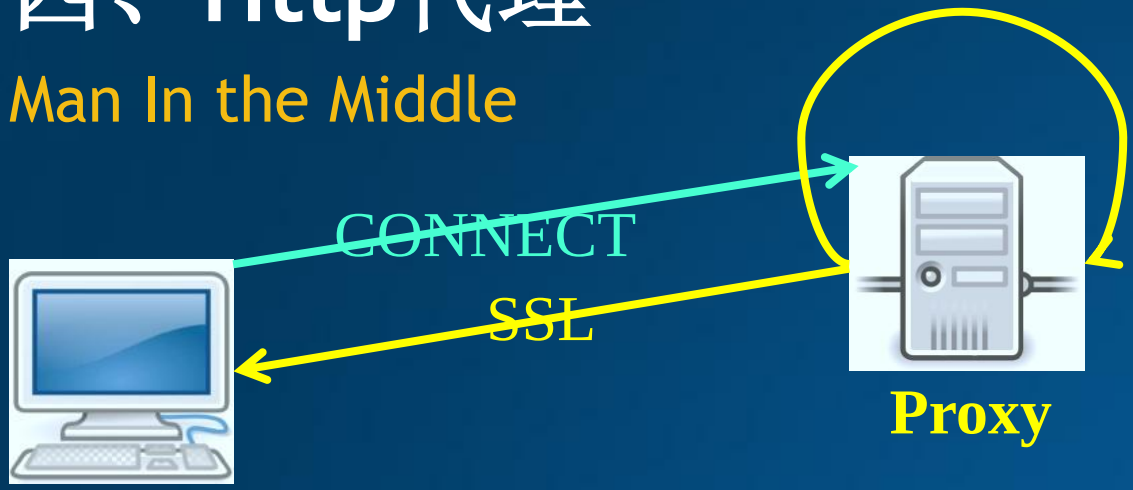


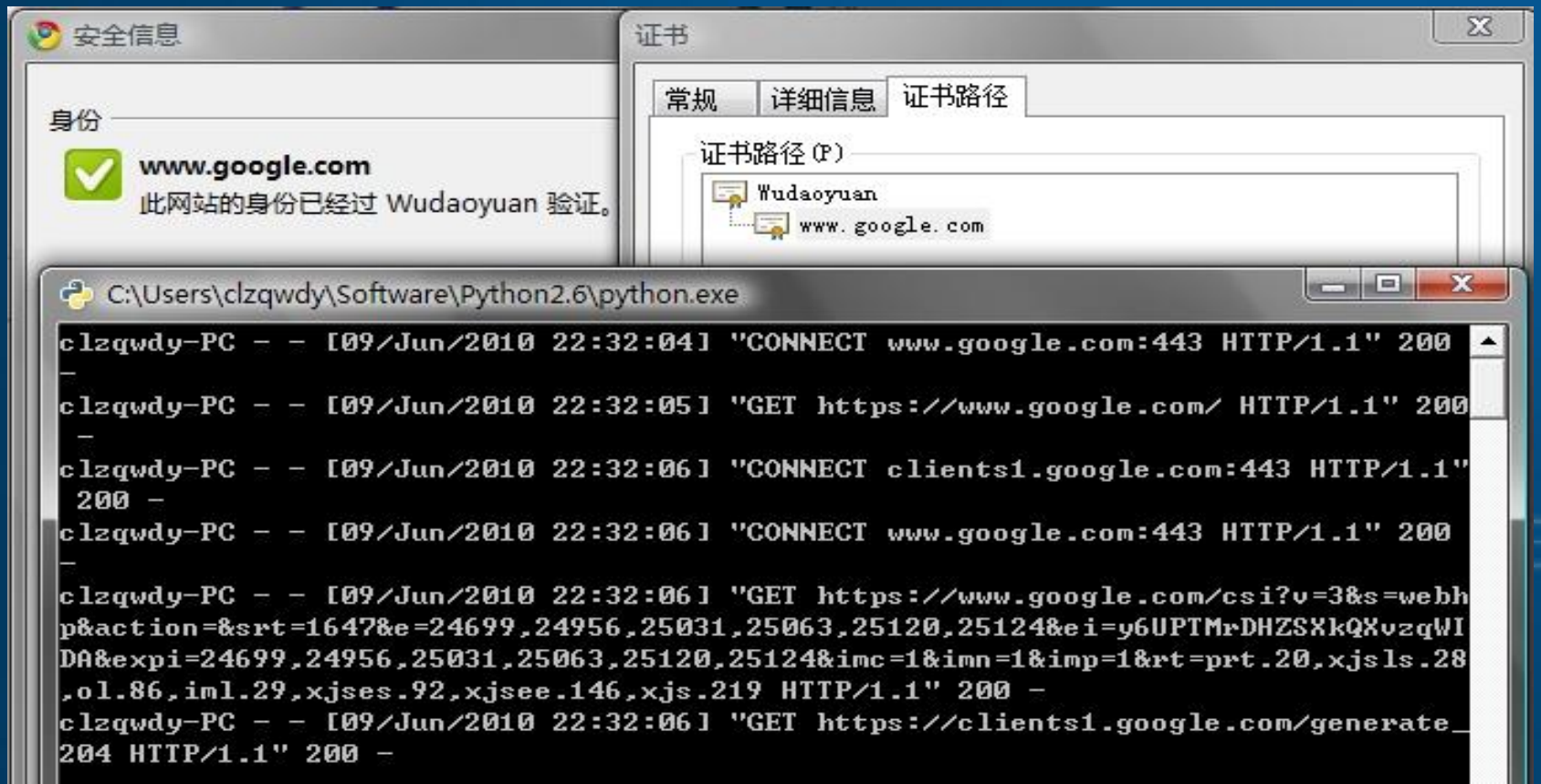


四、Http代理

CA sign a server cert to Host

Man In the Middle

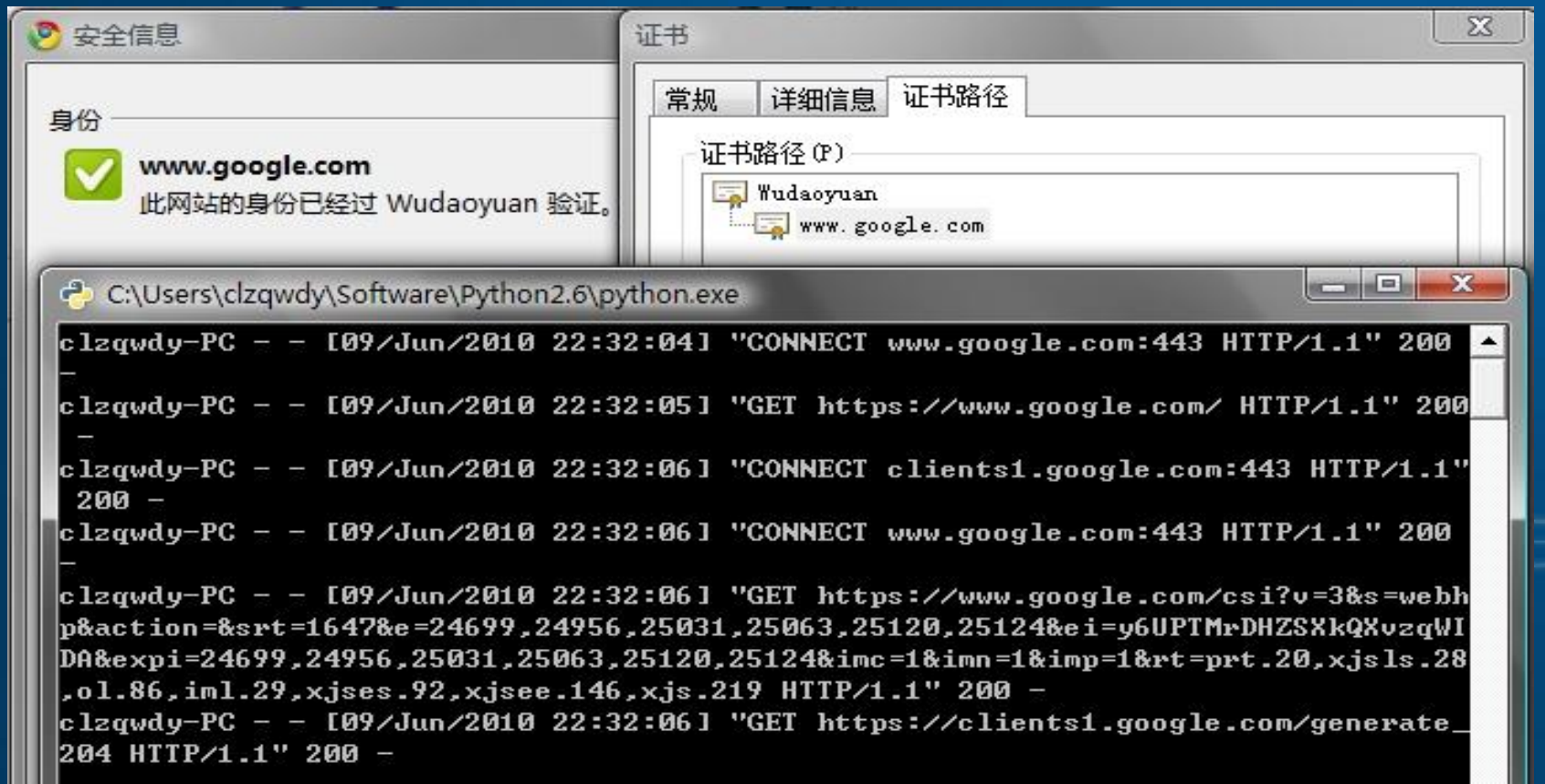




四、Http代理

CA sign a server cert to Host

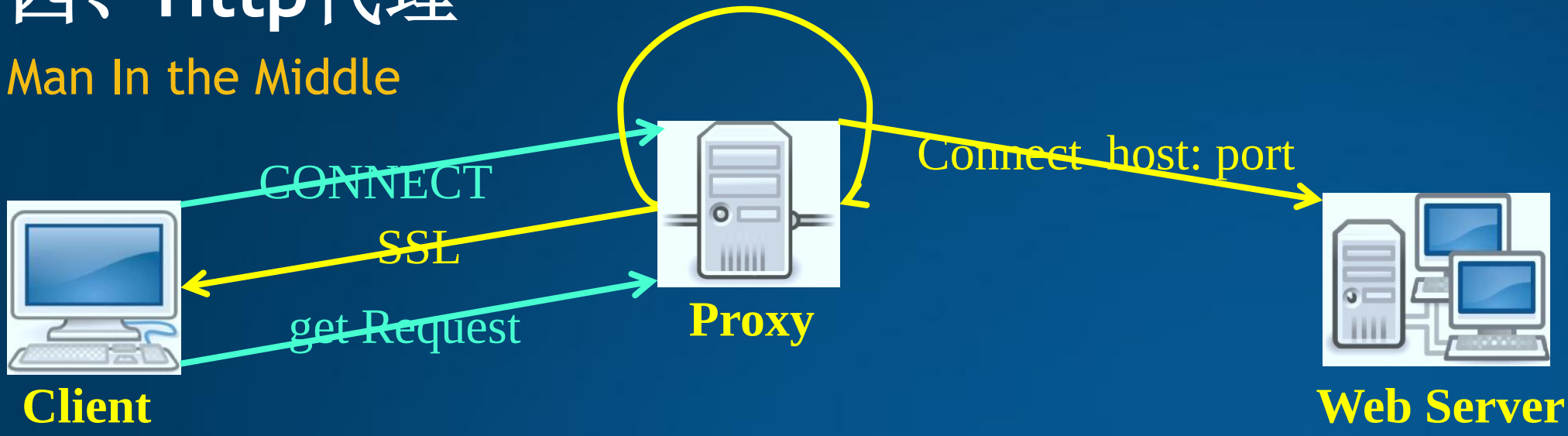
Man In the Middle

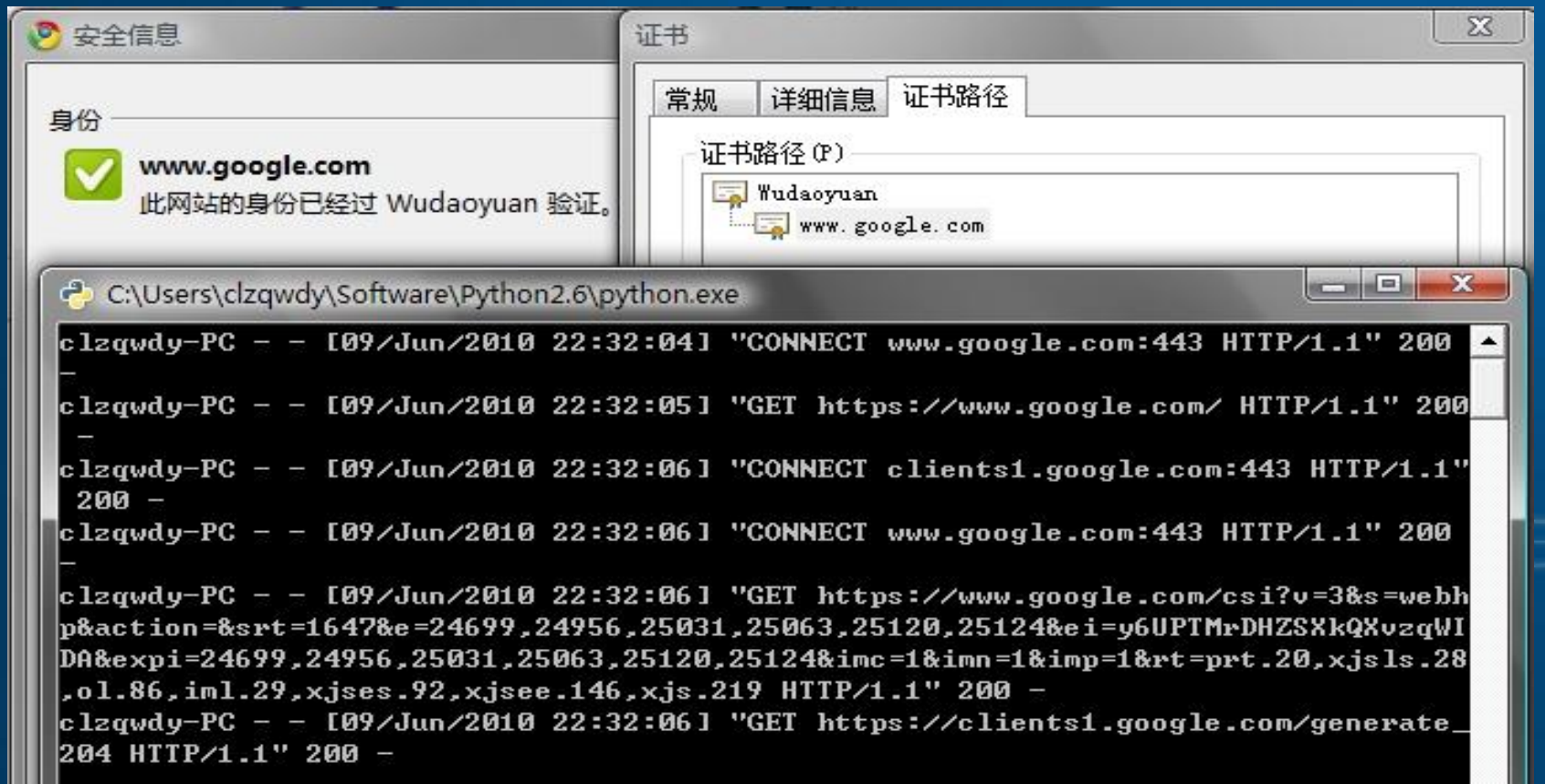


四、Http代理

CA sign a server cert to Host

Man In the Middle

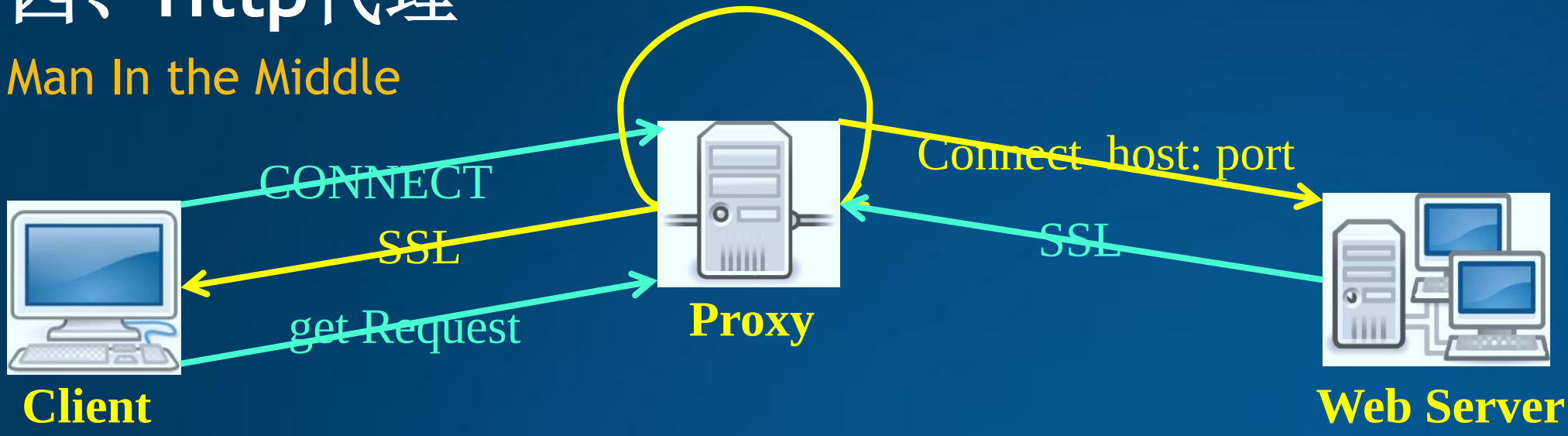


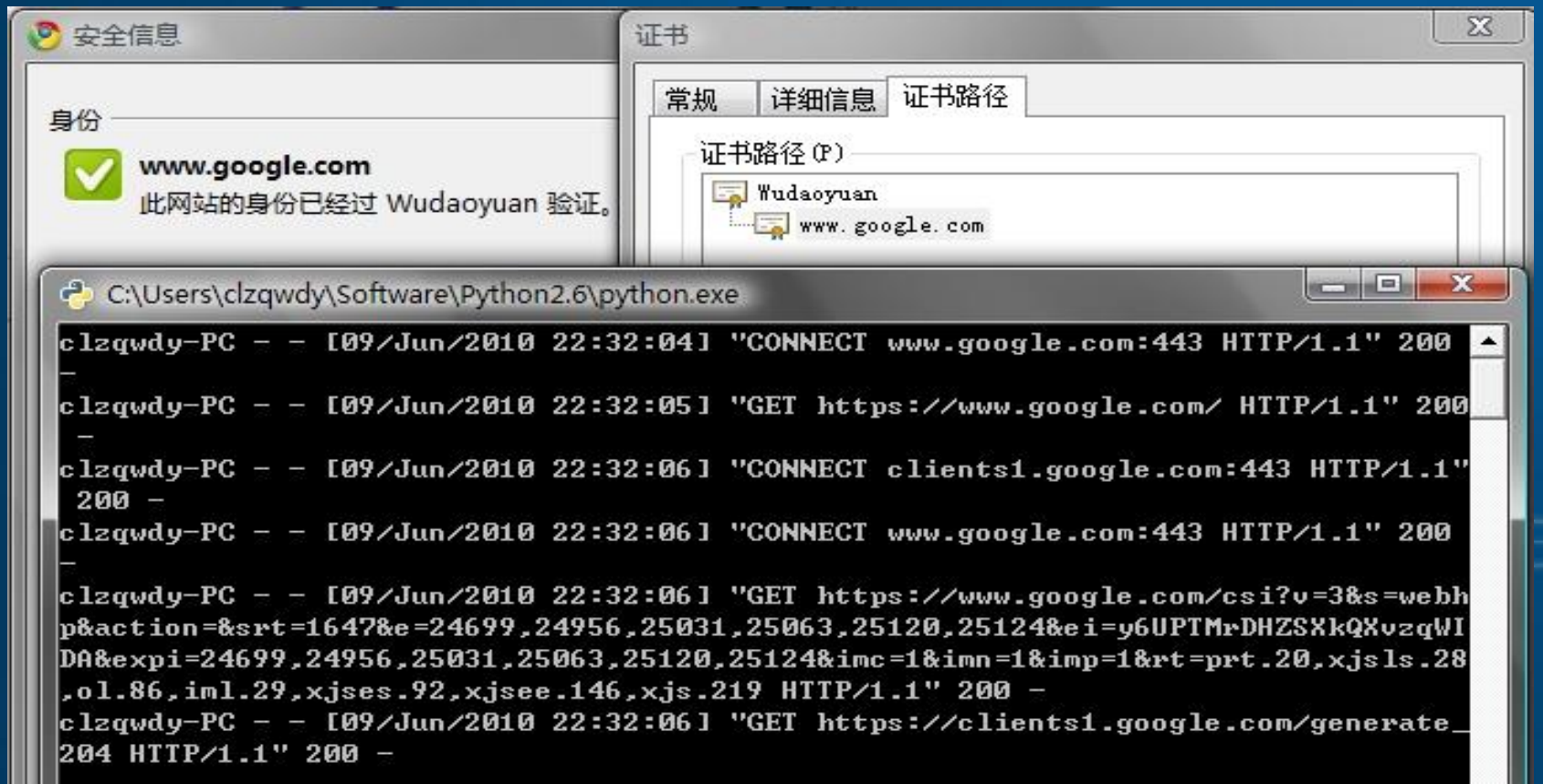


四、Http代理

CA sign a server cert to Host

Man In the Middle

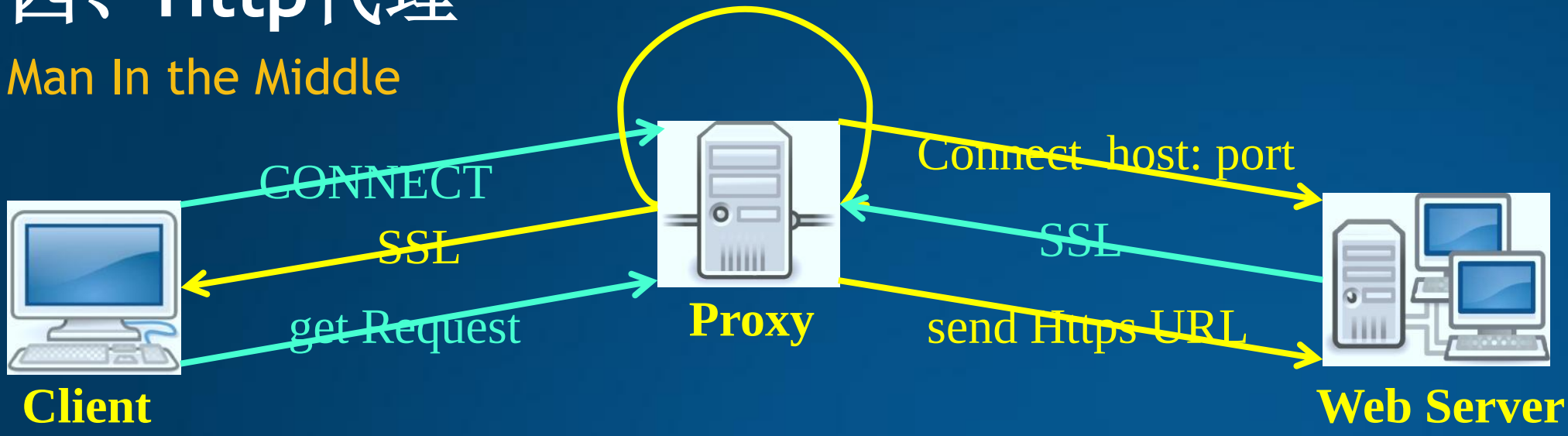


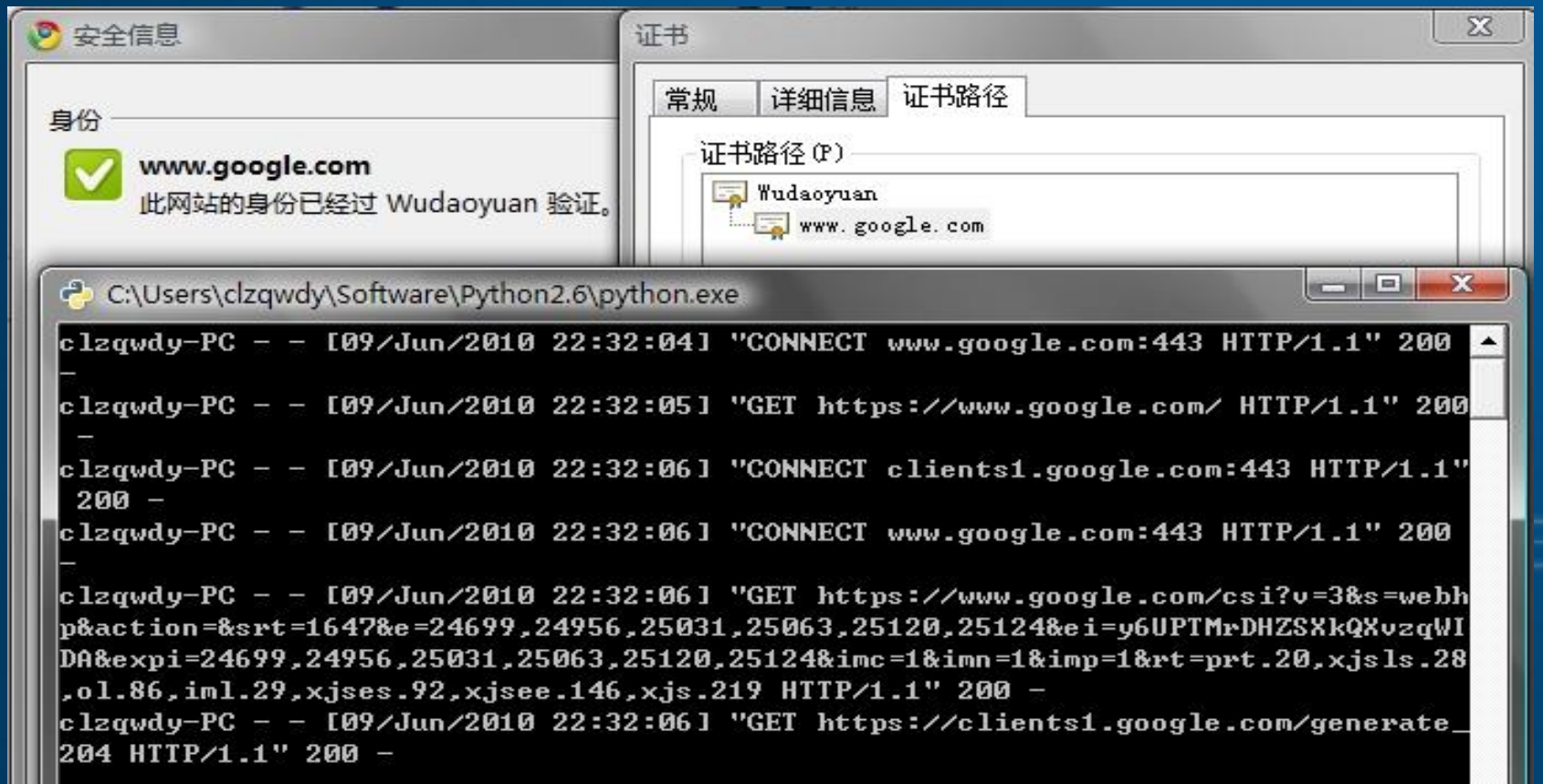


四、Http代理

CA sign a server cert to Host

Man In the Middle





Thank You Very Much!