奥联签名验签问题

首先我按照”403设备配置信息.xlsx“里的端口和ip对签名验签服务进行访问

名称	服务器类型	ip	网页地址及账号密码	ssh远程	备注
密码机	物理机	192.168.2.34		2222端口。	
统一密码服务器	虚机	192.168.2.38	超级管理员账号:super_admin 口令密码：olym_2023 机构管理员账号：xidian_admin 口令密码：olym_2023	账号:root 密码:admin@123	
vpn	虚机	192.168.2.40	网关登录链接：https://192.168.2.40:8443/admin 账号：admin 口令：olymntls_2023		
数字证书	虚机	192.168.2.42	https://192.168.2.42/olca/		
云密码机	物理机	192.168.2.43	192.168.2.43:8443　下挂密码机：192.168.2.44		
时间戳服务器（中安云科）	物理机	192.168.2.49			
sm9标识密码机	物理机				未使用
签名验签服务	物理机	192.168.2.45	https://192.168.2.45:8443/	Ukey登录	
IPsecvpn1	虚机	192.168.3.41	网关登录链接：https://192.168.2.40:8443/admin		
IPsecvpn2	虚机	192.168.4.41	账号：admin		

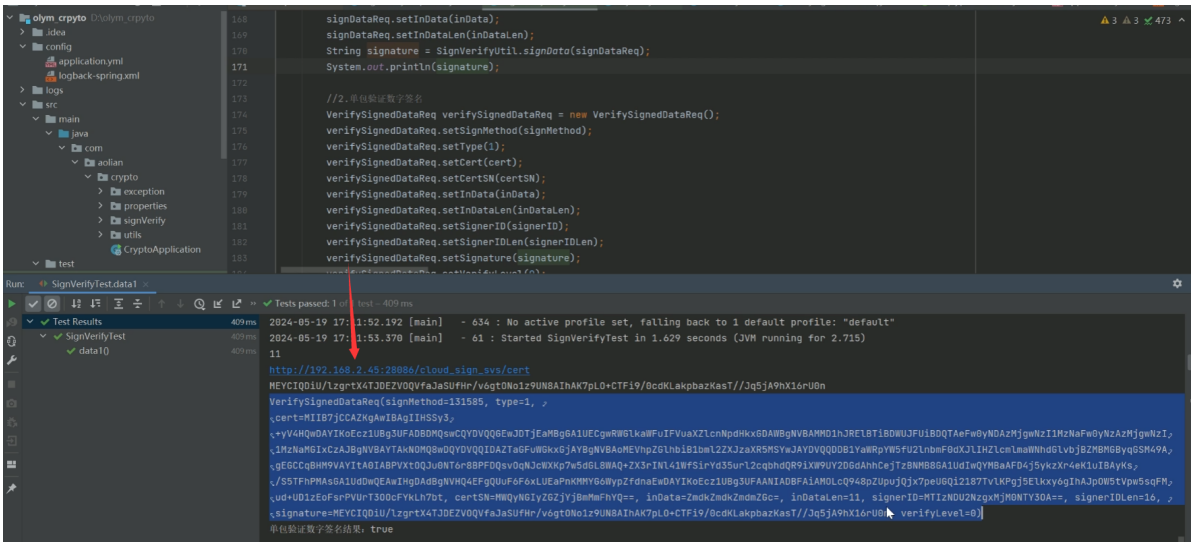
报错，没有https的证书，请求打不过去

```
curl: (60) Peer's Certificate issuer is not recognized.
More details here: http://curl.haxx.se/docs/sslcerts.html

curl performs SSL certificate verification by default, using a "bundle"
 of Certificate Authority (CA) public keys (CA certs). If the default
 bundle file isn't adequate, you can specify an alternate file
 using the --cacert option.
If this HTTPS server uses a certificate signed by a CA represented in
 the bundle, the certificate verification probably failed due to a
 problem with the certificate (it might be expired, or the name might
 not match the domain name in the URL).
If you'd like to turn off curl's verification of the certificate, use
 the -k (or --insecure) option.
```

然后我试着把https换成http，提示The plain HTTP request was sent to HTTPS port，还是打不过去

然后我看了那个视频，发现他在测试的时候用的并不是8443端口，而是28086（签名验签服务器的默认端口）



同时我试着ping了这台主机`192.168.2.45`，通了

```
[xzt@localhost ~]$ ping 192.168.2.45
PING 192.168.2.45 (192.168.2.45) 56(84) bytes of data.
64 bytes from 192.168.2.45: icmp_seq=1 ttl=64 time=0.118 ms
64 bytes from 192.168.2.45: icmp_seq=2 ttl=64 time=0.100 ms
64 bytes from 192.168.2.45: icmp_seq=3 ttl=64 time=0.106 ms
64 bytes from 192.168.2.45: icmp_seq=4 ttl=64 time=0.084 ms
64 bytes from 192.168.2.45: icmp_seq=5 ttl=64 time=0.113 ms
64 bytes from 192.168.2.45: icmp_seq=6 ttl=64 time=0.111 ms
64 bytes from 192.168.2.45: icmp_seq=7 ttl=64 time=0.113 ms
64 bytes from 192.168.2.45: icmp_seq=8 ttl=64 time=0.228 ms
64 bytes from 192.168.2.45: icmp_seq=9 ttl=64 time=0.105 ms

64 bytes from 192.168.2.45: icmp_seq=10 ttl=64 time=0.116 ms
^C
--- 192.168.2.45 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 8999ms
rtt min/avg/max/mdev = 0.084/0.119/0.228/0.038 ms
[xzt@localhost ~]$
```

于是我把端口换成28086，同时用的http（和测试视频中一样），发送签名请求仍然报错，端口API拒绝连接

```
进来了签名验签测试函数
初始化基本信息
初始化传参
发送请求
---------->验签出错
org.springframework.web.client.ResourceAccessException: I/O error on POST request for "http://192.168.2.45:28086/VerifySignedData": 拒绝连接 (Connection refused); nested exception is java.net.ConnectException: 拒绝连接 (Connection refused)
        at org.springframework.web.client.RestTemplate.doExecute(RestTemplate.java:784)
```

报错的代码行数如下图

```
30      // 单包数字签名
31  @   public static String signature(Map<String, Object> map) throws Exception {
32          // 初始化请求体
33          SVSRequest<SignDataReq> svsRequest = new SVSRequest<>();
34          svsRequest.setVersion(SVS_REQUEST_VERSION);
35          svsRequest.setReqType(ReqTypeConstants.REQTYPE_SIGNDATA);
36          svsRequest.setReqTime(DateUtil.generateSvsRequestTime());
37          // 解析传参，构造请求内容
38          // keyValue和keyIndex要从服务器上获得（写死的），表示私钥的内容和私钥的索引
39          // 私钥值
40          // 索引值
41          String inData = map.toString();
42          // 0x00020201 表示算法类型 SM3-SM2，根据国家密码标准
43          // KeyIndex 是什么东西？
44          SignDataReq signDataReq = new SignDataReq(SDG_SM3_SM2, keyIndex, keyValue, signerID.length(), signerID, inData.length(), in
45          svsRequest.setRequest(signDataReq);
46          headers.setContentType(MediaType.APPLICATION_JSON);
47
48          HttpEntity<?> requestEntity = new HttpEntity<>(svsRequest, headers);
49          ParameterizedTypeReference<SVSRespond<SignDataResp>> responseType =
50                  new ParameterizedTypeReference<SVSRespond<SignDataResp>>() {};
51
52          // 发送 POST 请求并获取响应
53          ResponseEntity<SVSRespond<SignDataResp>> responseEntity =
54                  restTemplate.exchange( url: signVerifyUrl + URIConstants.URI_SIGNDATA, HttpMethod.POST, requestEntity, responseType);
55
```

我感觉这个报错和参数是无关的，单纯这个http请求没打过去，被拒绝了，然后curl了一下这个api，同样是拒绝连接

```
[xzt@localhost ~]$ curl 192.168.2.45:28086/SignData
curl: (7) Failed connect to 192.168.2.45:28086; 拒绝连接
[xzt@localhost ~]$
```

然后我不知道怎么办了