软算法支持 SSL 卸载

部署使用说明

北京江南天安科技有限公司

2022 年 12 月

版本历史

文档版本	更新时间	修订人	文档更新说明
V1.00	2020-5-13	马晓艳	新建文档
V1.01	2021-01-21	闫世超	软算法修订版

目录

1. SSL 卸载业务场景

通过使用 TASSL，实现 SSL 卸载；全面支持国密算法证书和国密 SSL 协议，符合监管合规要求。

SSL 卸载业务的典型部署方案：

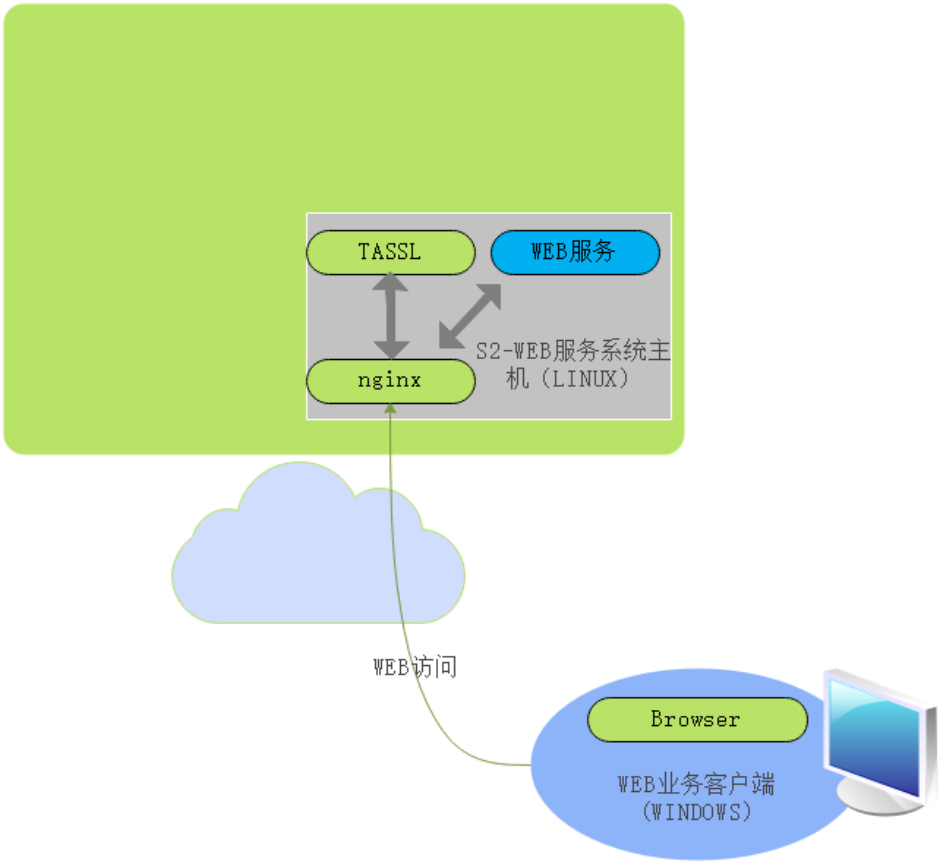


图 1 典型部署图

2. 所需资源

2.1. 硬件资源

如图 1 所示，用户所需的硬件资源环境如下：

类型	要求	说明
S2	Linux 64 位	用于部署用户的业务系统，TASSL 以及 nginx/httpd

防火墙策略		S2 对外提供 TLS 端口服务；

2.2. 软件资源

1. TASSL

   https://github.com/jntass/TASSL-1.1.1

2. nginx

   支持使用原生 nginx 配合 TASSL 构建基于国密 SSL 的 web server/反向代理，nginx 版本不应过低，本文档基于 nginx-1.14.2 测试

   https://nginx.org/en/download.html

3. httpd

   支持使用原生 httpd 配合 TASSL 构建基于国密 SSL 的 web server/反向代理，httpd 版本不应过低，本文档基于 httpd-2.4.46 测试

   https://httpd.apache.org/download.cgi

3. 主要步骤

3.1. TASSL 安装与测试

3.1.1 基于源码安装

1、 配置

```
./config --prefix=/usr/local/tassl no-shared
```

注：由于许多系统有自带的 ssl 库，为避免潜在的动态库冲突，此处仅生成静态库

2、 编译及安装

```
make && make install
```

3、 检查 TASSL 版本

```
/usr/local/tassl/bin/openssl version -a
```

```
[root@hsmyf_10 tassl]# /usr/local/tassl/bin/openssl version -a
OpenSSL 1.1.1s  Tassl 2.1.2 1 Nov 2022
built on: Tue Dec  6 09:52:46 2022 UTC
platform: linux-x86_64
options:  bn(64,64) rc4(16x,int) des(int) idea(int) blowfish(ptr)
compiler: gcc -fPIC -pthread -m64 -Wa,--noexecstack -Wall -O3 -DOPENSSL_USE_NODELETE -DL_ENDIAN -DOPENSSL_PIC -DOPENSSL_CPUID_OBJ -DOPENSSL_IA32_SSE2
 -DOPENSSL_BN_ASM_MONT -DOPENSSL_BN_ASM_MONT5 -DOPENSSL_BN_ASM_GF2m -DSHA1_ASM -DSHA256_ASM -DSHA512_ASM -DKECCAK1600_ASM -DRC4_ASM -DMD5_ASM -DAESNI_A
SM -DVPAES_ASM -DGHASH_ASM -DECP_NISTZ256_ASM -DX25519_ASM -DPOLY1305_ASM -DNDEBUG
OPENSSLDIR: "/usr/local/tassl/ssl"
ENGINESDIR: "/usr/local/tassl/lib/engines-1.1"
Seeding source: os-specific
[root@hsmyf_10 tassl]#
```

3.1.2 签发国密证书

按照实际业务需要产生 SM2 的证书申请文件并签发证书，测试阶段可以采用自签发证书，正式运行阶段建议从运营 CA 签发合格的服务器证书。

a)

cd /usr/local/tassl/tassl_demo/cert/

b)

./gen_sm2_cert.sh

注：执行脚本后当前目录下生成 certs 目录，CA.crt/CA.key 为根证书/密钥，SS.crt/SS.key 为服务端签名证书/密钥，SE.crt/SE.key 为服务端加密证书/密钥，CS.crt/CS.key 为客户端签名证书/密钥，CE.crt/CE.key 为客户端加密证书/密钥

3.1.3 s_server/s_client 测试国密 SSL

注:如使用 nginx/httpd 进行国密 SSL 卸载，可跳过此节。此节为使用 openssl 命令行 s_server/s_client 测试国密 SSL

a)

cd /usr/local/tassl/bin/

b) 启动服务端

./openssl s_server -cert ../tassl_demo/cert/certs/SS.crt -key ../tassl_demo/cert/certs/SS.key -cert_enc ../tassl_demo/cert/certs/SE.crt -key_enc ../tassl_demo/cert/certs/SE.key -accept 4433

c) 启动客户端

./openssl s_client -verify 10 -CAfile ../tassl_demo/cert/certs/CA.crt -connect 127.0.0.1:4433 -gmtls1_1

3.1.4 SSL demo 测试国密 SSL

注:如使用 nginx/httpd 进行国密 SSL 卸载，可跳过此节。此 demo 提供 SSL 服务端和客户端的最小示例工程，如需自行调用 TASSL 实现国密 SSL 功能，参考此 demo

a)

cd /usr/local/tassl/tassl_demo/ssl

b)

./mk.sh

c) 启动服务端

./sslsvr -p 4433 -sc ../cert/certs/SS.crt -sk ../cert/certs/SS.key -ec ../cert/certs/SE.crt -ek ../cert/certs/SE.key

```
[root@hsmyf_10 ssl]# ./sslsvr -p 4433 -sc ../cert/certs/SS.crt -sk ../cert/certs/SS.key -ec ../cert/certs/SE.crt -ek ../cert/certs/SE.
key
Start With
                Listening Port 4433
                Sign Cert ../cert/certs/SS.crt
                Sign Key ../cert/certs/SS.key
                Enc Cert ../cert/certs/SE.crt
                Enc Key ../cert/certs/SE.key
                CA Cert null
                Engine null
                Verify Peer False
```

其中：

-p：指定监听端口

-sc：指定 3.1.2 签发国密证书生成的签名证书

-sk：指定 3.1.2 签发国密证书生成的签名密钥

-ec：指定 3.1.2 签发国密证书生成的加密证书

-ek：指定 3.1.2 签发国密证书生成的加密密钥

d) 启动客户端

./sslcli -s 127.0.0.1:4433 --gmssl --verify -ca ../cert/certs/CA.crt

```
[root@hsmyf_10 ssl]# ./sslcli -s 127.0.0.1:4433 --gmssl --verify -ca ../cert/certs/CA.crt
Start With
                Server Address 127.0.0.1:4433
                Sign Cert null
                Sign Key null
                Enc Cert null
                Enc Key null
                CA Cert ../cert/certs/CA.crt
                Engine null
                Verify Peer True
SSL connection using GMTLSv1.1, ECC-SM4-SM3
对端证书信息:
证书: /C=CN/ST=BJ/L=HaiDian/O=Beijing JNTA Technology LTD./OU=BSRC of TASS/CN=server sign (SM2)
颁发者: /C=CN/ST=BJ/L=HaiDian/O=Beijing JNTA Technology LTD./OU=SORB of TASS/CN=Test CA (SM2)
recv 36 bytes : this message is from the SSL server!
[root@hsmyf_10 ssl]#
```

其中：

-s：指定服务器地址

--gmssl：使用国密 TLSv1.1

--verify：打开证书验证选项

-ca：指定 CA 证书

3.2. nginx 安装与测试

以下内容以 web server 方式介绍如何使用 nginx 搭建服务器并使用浏览器进行测试。

3.2.1 基于源码安装

1、 配置

./configure --without-http_uwsgi_module --with-http_ssl_module --with-stream --with-stream_ssl_module --prefix=/usr/local/nginx --with-openssl=/home/TASSL-1.1.1

注：--with-openssl 后跟 tassl 源码路径，非安装路径，以上配置会将 tassl 以静态库形式编译进 nginx 中

```
checking for struct dirent.d_namlen ... not found
checking for struct dirent.d_type ... found
checking for sysconf(_SC_NPROCESSORS_ONLN) ... found
checking for sysconf(_SC_LEVEL1_DCACHE_LINESIZE) ... found
checking for openat(), fstatat() ... found
checking for getaddrinfo() ... found
checking for PCRE library ... found
checking for PCRE JIT support ... not found
checking for zlib library ... found
creating objs/Makefile

Configuration summary
  + using system PCRE library
  + using OpenSSL library: /home/TASSL-1.1.1
  + using system zlib library

  nginx path prefix: "/usr/local/nginx"
  nginx binary file: "/usr/local/nginx/sbin/nginx"
  nginx modules path: "/usr/local/nginx/modules"
  nginx configuration prefix: "/usr/local/nginx/conf"
  nginx configuration file: "/usr/local/nginx/conf/nginx.conf"
  nginx pid file: "/usr/local/nginx/logs/nginx.pid"
  nginx error log file: "/usr/local/nginx/logs/error.log"
  nginx http access log file: "/usr/local/nginx/logs/access.log"
  nginx http client request body temporary files: "client_body_temp"
  nginx http proxy temporary files: "proxy_temp"
  nginx http fastcgi temporary files: "fastcgi_temp"
  nginx http scgi temporary files: "scgi_temp"

[root@hsmyf_10 nginx-1.14.2]#
```

2、 编译及安装

make && make install

```
                || cp conf/mime.types '/usr/local/nginx/conf'
cp conf/mime.types '/usr/local/nginx/conf/mime.types.default'
test -f '/usr/local/nginx/conf/fastcgi_params' \
                || cp conf/fastcgi_params '/usr/local/nginx/conf'
cp conf/fastcgi_params \
                '/usr/local/nginx/conf/fastcgi_params.default'
test -f '/usr/local/nginx/conf/fastcgi.conf' \
                || cp conf/fastcgi.conf '/usr/local/nginx/conf'
cp conf/fastcgi.conf '/usr/local/nginx/conf/fastcgi.conf.default'
test -f '/usr/local/nginx/conf/uwsgi_params' \
                || cp conf/uwsgi_params '/usr/local/nginx/conf'
cp conf/uwsgi_params \
                '/usr/local/nginx/conf/uwsgi_params.default'
test -f '/usr/local/nginx/conf/scgi_params' \
                || cp conf/scgi_params '/usr/local/nginx/conf'
cp conf/scgi_params \
                '/usr/local/nginx/conf/scgi_params.default'
test -f '/usr/local/nginx/conf/nginx.conf' \
                || cp conf/nginx.conf '/usr/local/nginx/conf/nginx.conf'
cp conf/nginx.conf '/usr/local/nginx/conf/nginx.conf.default'
test -d '/usr/local/nginx/logs' \
                || mkdir -p '/usr/local/nginx/logs'
test -d '/usr/local/nginx/logs' \
                || mkdir -p '/usr/local/nginx/logs'
test -d '/usr/local/nginx/html' \
                || cp -R html '/usr/local/nginx'
test -d '/usr/local/nginx/logs' \
                || mkdir -p '/usr/local/nginx/logs'
make[1]: Leaving directory `/root/nginx-1.14.2'
[root@hsmyf_10 nginx-1.14.2]#
```

3、 检查 nginx 版本

/usr/local/nginx/sbin/nginx -V

```
[root@hsmyf_10 nginx-1.14.2]# /usr/local/nginx/sbin/nginx -V
nginx version: nginx/1.14.2
built by gcc 4.4.7 20120313 (Red Hat 4.4.7-17) (GCC)
built with OpenSSL 1.1.1s  Tassl 2.1.2 1 Nov 2022
TLS SNI support enabled
configure arguments: --without-http_uwsgi_module --with-http_ssl_module --with-stream --with-stream_ssl_module --prefix=/usr/local/nginx --with-open
ssl=/home/TASSL-1.1.1
[root@hsmyf_10 nginx-1.14.2]#
```

3.2.2 修改并测试配置文件

1、 使用国密双证书

a)

cd /usr/local/nginx/conf

b) 修改 nginx.conf 为以下内容

```
user  root;
worker_processes  auto;

events {
    worker_connections  1024;
}

http {
    include       mime.types;
    default_type  application/octet-stream;

    server {
        listen       8020 ssl;
        server_name  localhost;

        #not use engine
        ssl_certificate          /usr/local/tassl/tassl_demo/cert/certs/SS.crt;
        ssl_certificate_key      /usr/local/tassl/tassl_demo/cert/certs/SS.key;
        ssl_certificate          /usr/local/tassl/tassl_demo/cert/certs/SE.crt;
        ssl_certificate_key      /usr/local/tassl/tassl_demo/cert/certs/SE.key;

        location / {
            root   html;
            index  index.html index.htm;
        }

        error_page   500 502 503 504  /50x.html;
        location = /50x.html {
            root   html;
        }
    }
}
```

其中：

ssl_certificate：为 3.1.2 签发国密证书生成的签名证书

ssl_certificate_key：为 3.1.2 签发国密证书生成的签名密钥

ssl_certificate：为 3.1.2 签发国密证书生成的加密证书

ssl_certificate_key：为 3.1.2 签发国密证书生成的加密密钥

c)

/usr/local/nginx/sbin/nginx -t

```
[root@hsmyf_10 conf]# /usr/local/nginx/sbin/nginx -t
nginx: the configuration file /usr/local/nginx/conf/nginx.conf syntax is ok
nginx: configuration file /usr/local/nginx/conf/nginx.conf test is successful
[root@hsmyf_10 conf]#
```

3.2.3 启动 nginx

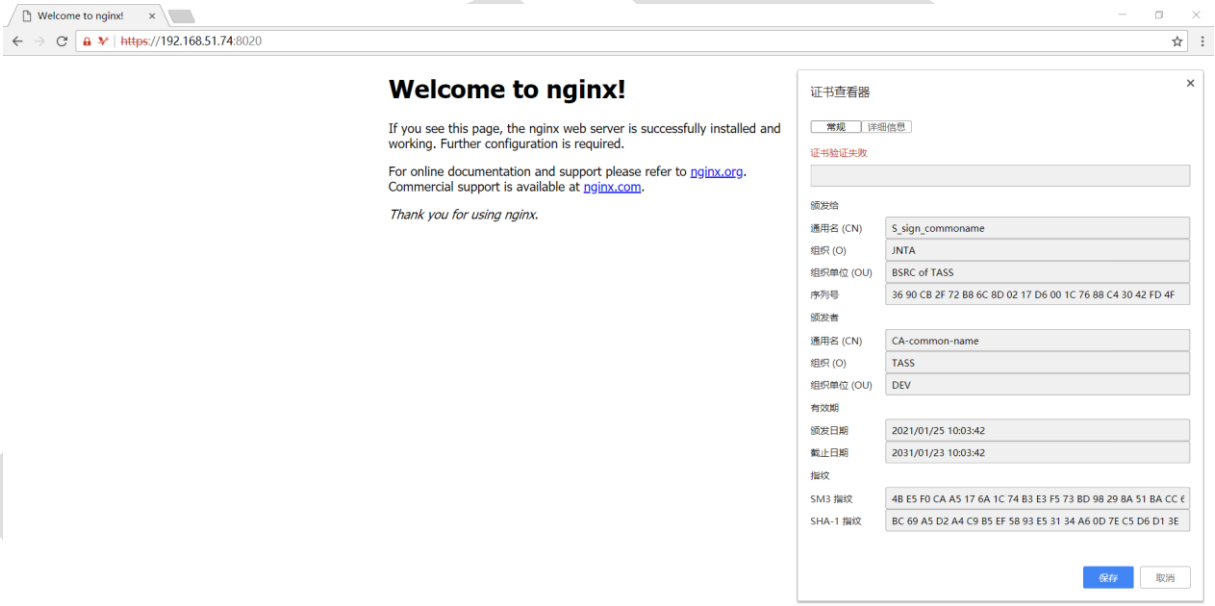
a)

/usr/local/nginx/sbin/nginx

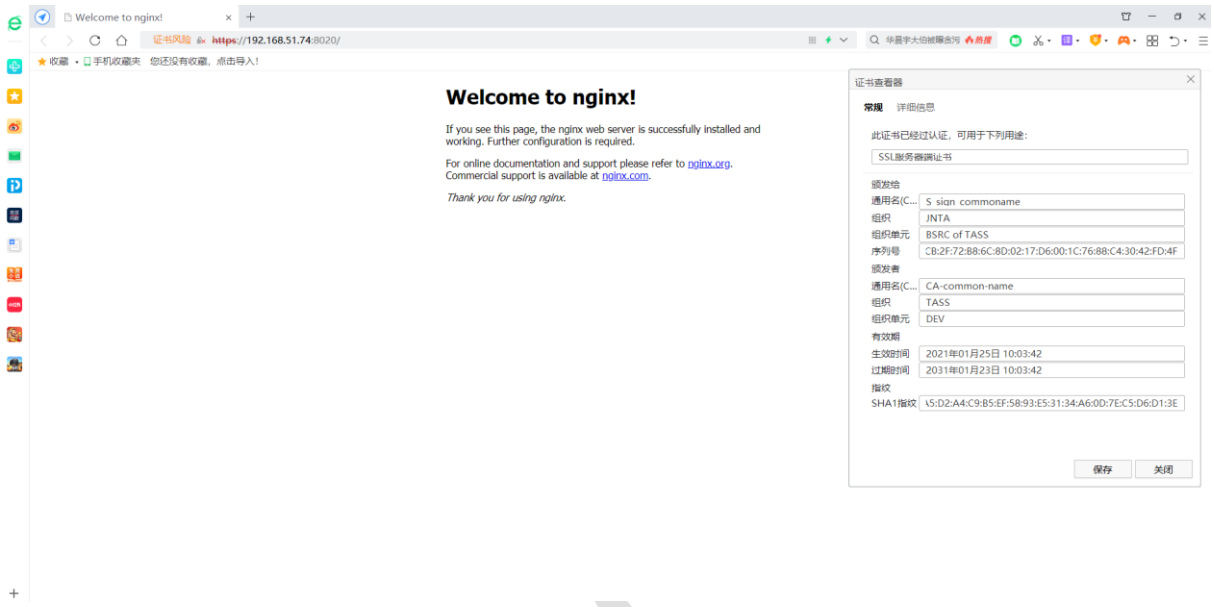
```
[root@hsmyf_10 conf]# /usr/local/nginx/sbin/nginx
[root@hsmyf_10 conf]#
```

3.2.4 测试验证 nginx

a) 密信浏览器



b) 360 浏览器



注：360 浏览器需启用国密 SSL 协议支持

3.3. httpd 安装与测试

以下内容以 web server 方式介绍如何使用 httpd 搭建服务器并使用浏览器进行测试。

3.3.1 基于源码安装

1、 配置

./configure --prefix=/usr/local/apache --with-apr=/usr/local/apr --with-apr-util=/usr/local/apr-util --enable-ssl --enable-ssl-staticlib-deps --with-ssl=/usr/local/tassl

注：--with-ssl 后跟 tassl 安装路径，以上配置会将 tassl 以静态库形式编译进模块 mod_ssl.so 中；安装 httpd 前请自行安装 expat/apr/apr-util 等依赖库

```
config.status: creating build/rules.mk
config.status: creating build/pkg/pkginfo
config.status: creating build/config_vars.sh
config.status: creating include/ap_config_auto.h
config.status: include/ap_config_auto.h is unchanged
config.status: executing default commands
configure: summary of build options:

    Server Version: 2.4.46
    Install prefix: /usr/local/apache
    C compiler:     gcc -std=gnu99
    CFLAGS:          -g -O2 -pthread
    CPPFLAGS:        -DLINUX -D_REENTRANT -D_GNU_SOURCE
    LDFLAGS:
    LIBS:
    C preprocessor: gcc -E

[root@hsmyf_10 httpd-2.4.46]#
```

2、 编译及安装

make && make install

```
Installing build system files
mkdir /usr/local/apache/build
Installing man pages and online manual
mkdir /usr/local/apache/man
mkdir /usr/local/apache/man/man1
mkdir /usr/local/apache/man/man8
mkdir /usr/local/apache/manual
make[1]: Leaving directory `/root/httpd-2.4.46'
[root@hsmyf_10 httpd-2.4.46]#
```

3、 检查 httpd 版本

/usr/local/apache/bin/apachectl -V

```
[root@hsmyf_10 httpd-2.4.46]# /usr/local/apache/bin/apachectl -V
Server version: Apache/2.4.46 (Unix)
Server built:   Dec  6 2022 19:32:19
Server's Module Magic Number: 20120211:93
Server loaded:  APR 1.5.2, APR-UTIL 1.6.1
Compiled using: APR 1.5.2, APR-UTIL 1.6.1
Architecture:   64-bit
Server MPM:     event
  threaded:     yes (fixed thread count)
    forked:     yes (variable process count)
Server compiled with....
 -D APR_HAS_SENDFILE
 -D APR_HAS_MMAP
 -D APR_HAVE_IPV6 (IPv4-mapped addresses enabled)
 -D APR_USE_SYSVSEM_SERIALIZE
 -D APR_USE_PTHREAD_SERIALIZE
 -D SINGLE_LISTEN_UNSERIALIZED_ACCEPT
 -D APR_HAS_OTHER_CHILD
 -D AP_HAVE_RELIABLE_PIPED_LOGS
 -D DYNAMIC_MODULE_LIMIT=256
 -D HTTPD_ROOT="/usr/local/apache"
 -D SUEXEC_BIN="/usr/local/apache/bin/suexec"
 -D DEFAULT_PIDLOG="logs/httpd.pid"
 -D DEFAULT_SCOREBOARD="logs/apache_runtime_status"
 -D DEFAULT_ERRORLOG="logs/error_log"
 -D AP_TYPES_CONFIG_FILE="conf/mime.types"
 -D SERVER_CONFIG_FILE="conf/httpd.conf"
[root@hsmyf_10 httpd-2.4.46]#
```

3.3.2 修改并测试配置文件

1、 使用国密双证书

a)

cd /usr/local/apache/conf

b) 修改 httpd.conf

```
84 #LoadModule file_cache_module modules/mod_file_cache.so
85 #LoadModule cache_module modules/mod_cache.so
86 #LoadModule cache_disk_module modules/mod_cache_disk.so
87 #LoadModule cache_socache_module modules/mod_cache_socache.so
88 LoadModule socache_shmcb_module modules/mod_socache_shmcb.so
89 #LoadModule socache_dbm_module modules/mod_socache_dbm.so
90 #LoadModule socache_memcache_module modules/mod_socache_memcache.so
91 #LoadModule socache_redis_module modules/mod_socache_redis.so
```

```
#LoadModule session_dbd_module modules/mod_session_dbd.so
#LoadModule slotmem_shm_module modules/mod_slotmem_shm.so
LoadModule ssl_module modules/mod_ssl.so
#LoadModule lbmethod_byrequests_module modules/mod_lbmethod_byrequests.so
#LoadModule lbmethod_bytraffic_module modules/mod_lbmethod_bytraffic.so
#LoadModule lbmethod_bybusyness_module modules/mod_lbmethod_bybusyness.so
#LoadModule lbmethod_heartbeat_module modules/mod_lbmethod_heartbeat.so
LoadModule unixd_module modules/mod_unixd.so
#LoadModule dav_module modules/mod_dav.so
LoadModule status_module modules/mod_status.so
LoadModule autoindex_module modules/mod_autoindex.so
```

```
# Secure (SSL/TLS) connections
Include conf/extra/httpd-ssl.conf
#
# Note: The following must must be present to support
#       starting without SSL on platforms with no /dev/random equivalent
#       but a statically compiled-in mod_ssl.
#
<IfModule ssl_module>
SSLRandomSeed startup builtin
SSLRandomSeed connect builtin
</IfModule>

```

c)

cd /usr/local/apache/conf/extra

d) 修改 httpd-ssl.conf

```
#   Some ECC cipher suites (http://www.ietf.org/rfc/rfc4492.txt)
#   require an ECC certificate which can also be configured in
#   parallel.
SSLCertificateFile "/usr/local/tassl/tassl_demo/cert/certs/SS.crt"
SSLCertificateFile "/usr/local/tassl/tassl_demo/cert/certs/SE.crt"
#SSLCertificateFile "/usr/local/apache/conf/server-dsa.crt"
#SSLCertificateFile "/usr/local/apache/conf/server-ecc.crt"

#   Server Private Key:
#   If the key is not combined with the certificate, use this
#   directive to point at the key file.  Keep in mind that if
#   you've both a RSA and a DSA private key you can configure
#   both in parallel (to also allow the use of DSA ciphers, etc.)
#   ECC keys, when in use, can also be configured in parallel
SSLCertificateKeyFile "/usr/local/tassl/tassl_demo/cert/certs/SS.key"
SSLCertificateKeyFile "/usr/local/tassl/tassl_demo/cert/certs/SE.key"
#SSLCertificateKeyFile "/usr/local/apache/conf/server-dsa.key"
#SSLCertificateKeyFile "/usr/local/apache/conf/server-ecc.key"
```

其中：

SSLCertificateFile：为 3.1.2 签发国密证书生成的签名证书

SSLCertificateFile：为 3.1.2 签发国密证书生成的加密证书

SSLCertificateKeyFile：为 3.1.2 签发国密证书生成的签名密钥

SSLCertificateKeyFile：为 3.1.2 签发国密证书生成的加密密钥

c)

/usr/local/apache/bin/apachectl -t

```
[root@hsmyf_10 bin]# /usr/local/apache/bin/apachectl -t
Syntax OK
[root@hsmyf_10 bin]#
```

3.3.3 启动 httpd

a)

/usr/local/apache/bin/apachectl start

```
[root@hsmyf_10 bin]# /usr/local/apache/bin/apachectl start
[root@hsmyf_10 bin]#
```

3.3.4 测试验证 httpd

a) 密信浏览器

b) 360 浏览器

注：360 浏览器需启用国密 SSL 协议支持

3.4. 常见问题

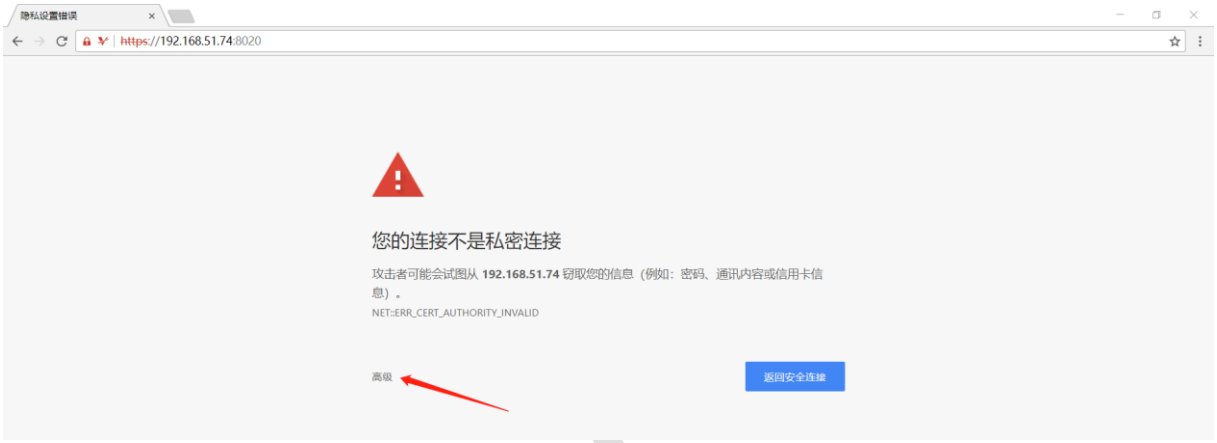
Q：360 浏览器测试国密 SSL，无法访问

A：确认是否”启用国密 SSL 协议支持”

Q：360 浏览器测试国密 SSL，显示服务器证书验证出错

A：切换”急速模式”

Q：浏览器测试验证，显示如下问题



A：证书不受信，选择高级->继续访问

Q：浏览器测试验证，显示证书风险

A：证书不受信；可将证书导入到浏览器可信证书或从受信 CA 签发证书