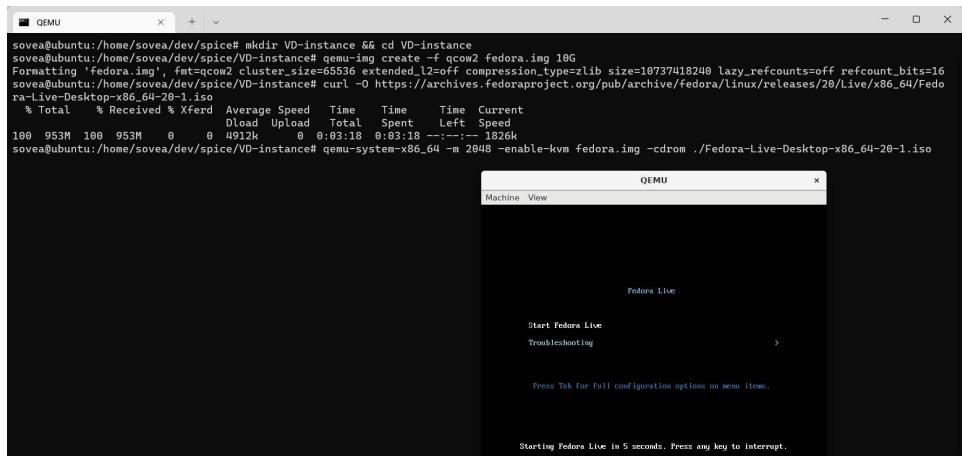


实验测试和分析

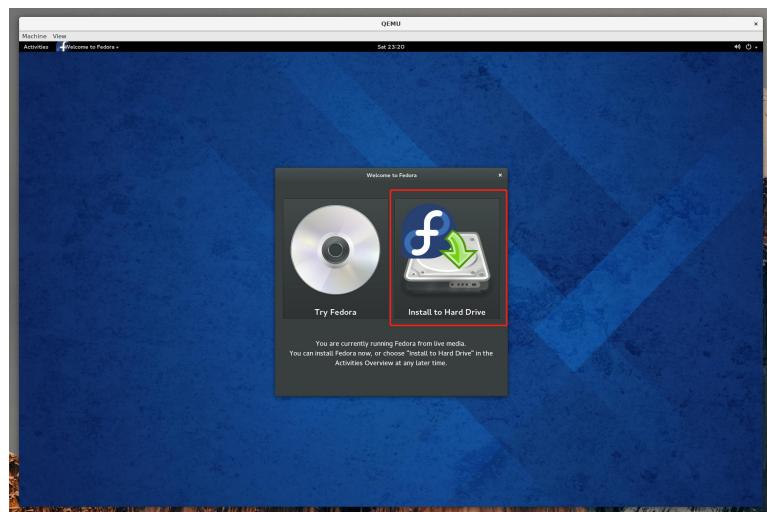
依照上述方案进行实现，最终完成了涵盖 C/S 及 B/S 架构的 Spice 协议改造。本节将利用 QEMU 创建并安装 Fedora 虚拟机，开启 Spice 传输功能，过程分为 C/S 与 B/S 两部分，利用 Spicy 及 Spice-html5 测试连接。通过使用 QEMU 创建的虚拟机进行连接测试以及传输效率对比，得到方案实施及分析结果。

QEMU 安装测试镜像

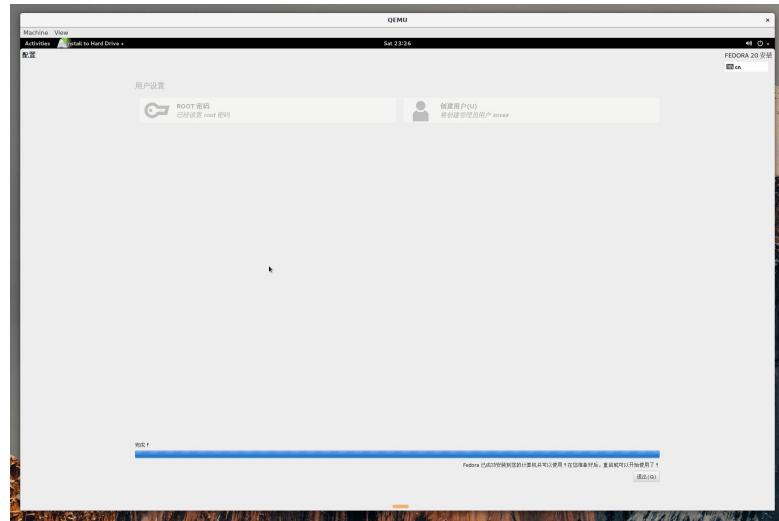
首先利用 qemu-img 指定大小创建虚拟机镜像，并下载 Fedora 系统镜像。利用 qemu-system-x86_64 启动并为虚拟机安装系统。过程如图 1(a) - 1(d)所示。



(a)



(b)



(c)



(d)

图 1 QEMU 虚拟机安装

C/S 测试

C/S 测试主要包括 Server、Client 加密传输及口令身份验证，具体是测试基于国密及非国密的 TLS 传输，以及基于 RSA/SM2 的口令身份验证，以此证明 C/S 架构下改造方案可行，并保持双方案兼容共存。

证书签发

首先签发符合 CA->Server 证书链的 RSA 及 SM2 单证书，如下图 2 所示，相关命令已整合成为 shell 脚本。

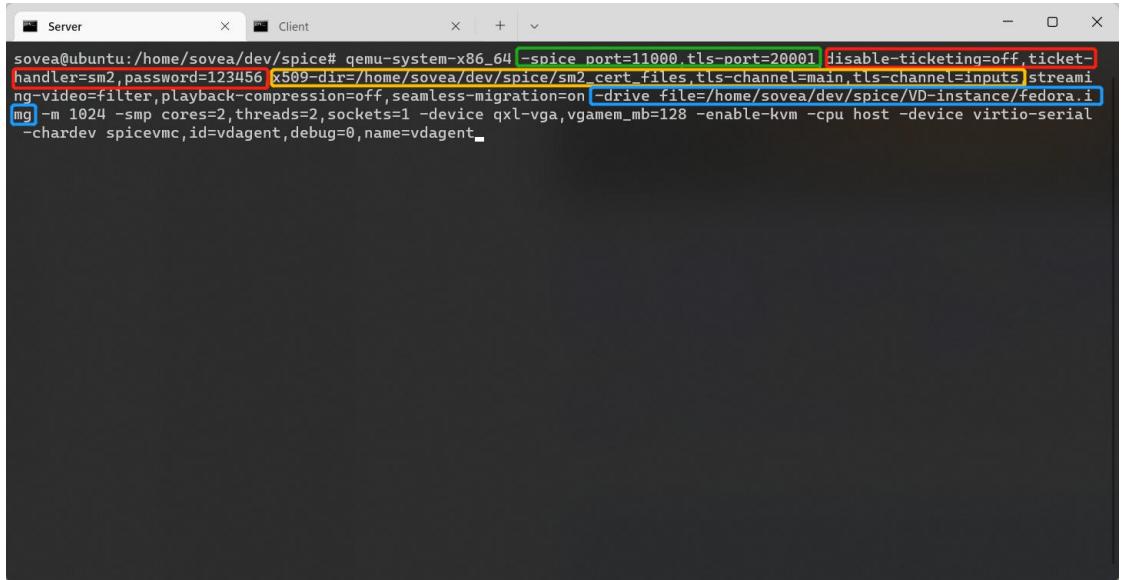


```
sovea@ubuntu:/home/sovea/dev/spice# ls rsa_cert_files/
ca-cert.pem ca-key.pem gen_certs.sh server-cert.pem server-key.csr server-key.pem server-key.pem.secure
sovea@ubuntu:/home/sovea/dev/spice# ls sm2_cert_files/
ca-cert.pem ca-key.csr ca-key.pem gen_certs.sh server-cert.pem server-key.csr server-key.pem
sovea@ubuntu:/home/sovea/dev/spice#
```

图 2 RSA 及 SM2 单证书

基于国密的 TLS 传输及 SM2 口令身份验证

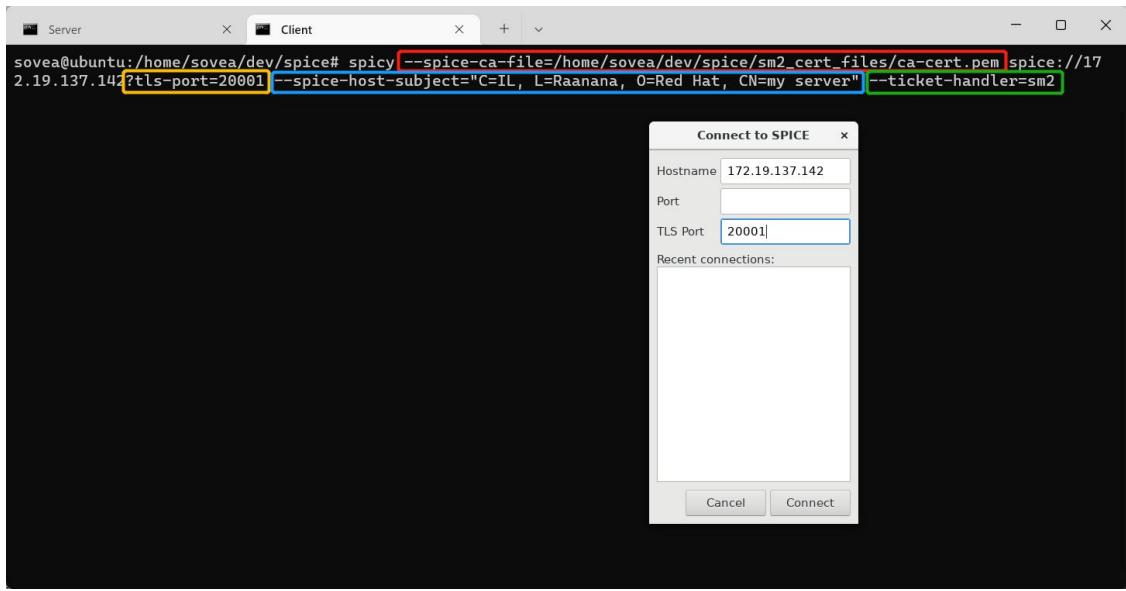
利用 qemu-system-x86_64 启动虚拟机并开启 Spice 传输，利用 SM2 单证书进行国密传输，并指定 ticket-handler 参数为 sm2(默认)在口令验证时使用 SM2 算法。设置默认端口及 tls 端口，开启口令验证，设置 password，x509-dir 指定 SM2 单证书所在目录，Spice 将据此读取所需 CA、Server 证书，tls-channel 指定需加密通道，运行前文所述安装的 Fedora 虚拟机，如图 3 所示。



```
sovea@ubuntu:/home/sovea/dev/spice# qemu-system-x86_64 -spice port=11000,tls-port=20001 disable-ticketing=off ticket-handler=sm2,password=123456 x509-dir=/home/sovea/dev/spice/sm2_cert_files,tls-channel=main,tls-channel=inputs streamlining-video=filter,playback-compression=off,seamless-migration=on -drive file=/home/sovea/dev/spice/VD-instance/fedora.img -m 1024 -smp cores=2,threads=2,sockets=1 -device qxl-vga,vgamem_mb=128 -enable-kvm -cpu host -device virtio-serial -chardev spicevnc,id=vdagent,debug=0,name=vdagent
```

图 3 C/S 测试 QEMU 启用 Spice 国密传输及 SM2 口令验证

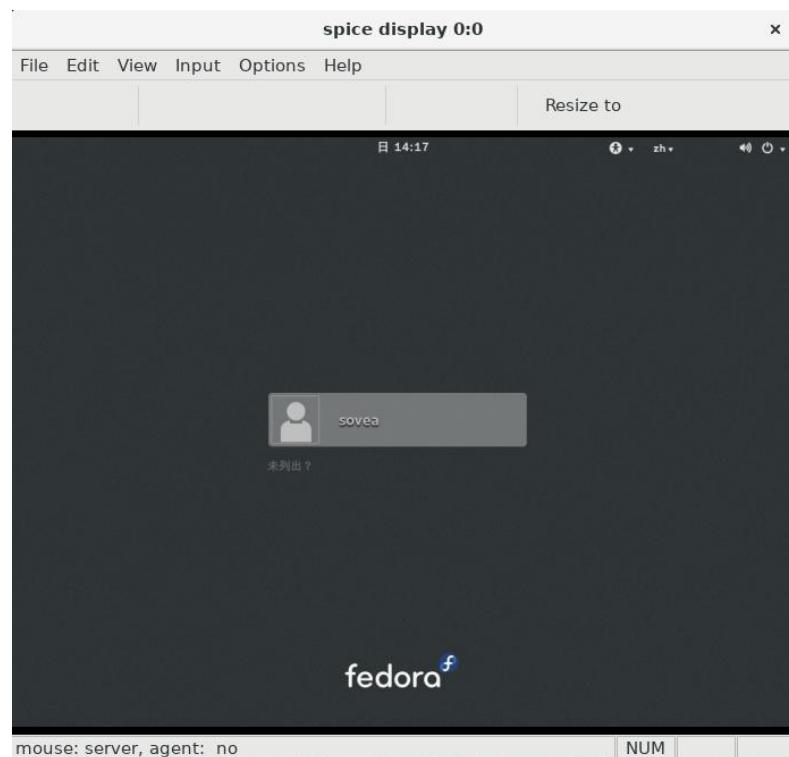
Spicy 客户端指定 CA 证书、连接地址及端口、证书主题字段进行连接，同时指定参数 ticket-handler 为 sm2(默认)，使用 SM2 算法进行口令验证。输入 password 通过验证完成连接，显示远程虚拟机桌面。过程如下图 4(a) - 4(c)所示。



(a)



(b)



(c)

图 4 Spicy 使用国密证书发起连接及 SM2 口令验证

连接过程中使用 tcpdump 对 Server 监听 tls-port 进行抓包，利用 Wireshark 进行分析可查看当前 TLS 传输连接的密钥协商结果，从而验证国密改造完成情况，如下图 5、6 所示。根据 Wireshark 分析可知连接使用 Cipher Suite 为 0x00c6，与前文 RFC 8998 新增国密套件 TLS_SM4_GCM_SM3 一致，证明当前 C/S 国密改造方案有效。

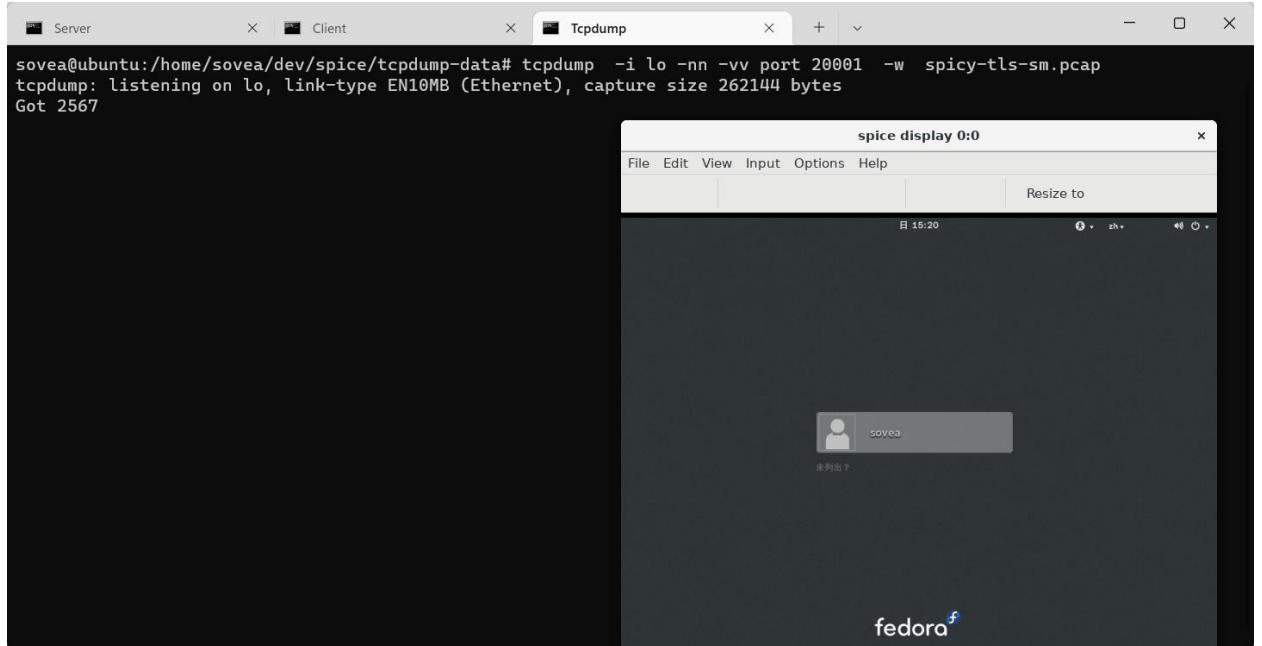


图 5 tcpdump 抓取 C/S 国密连接数据包

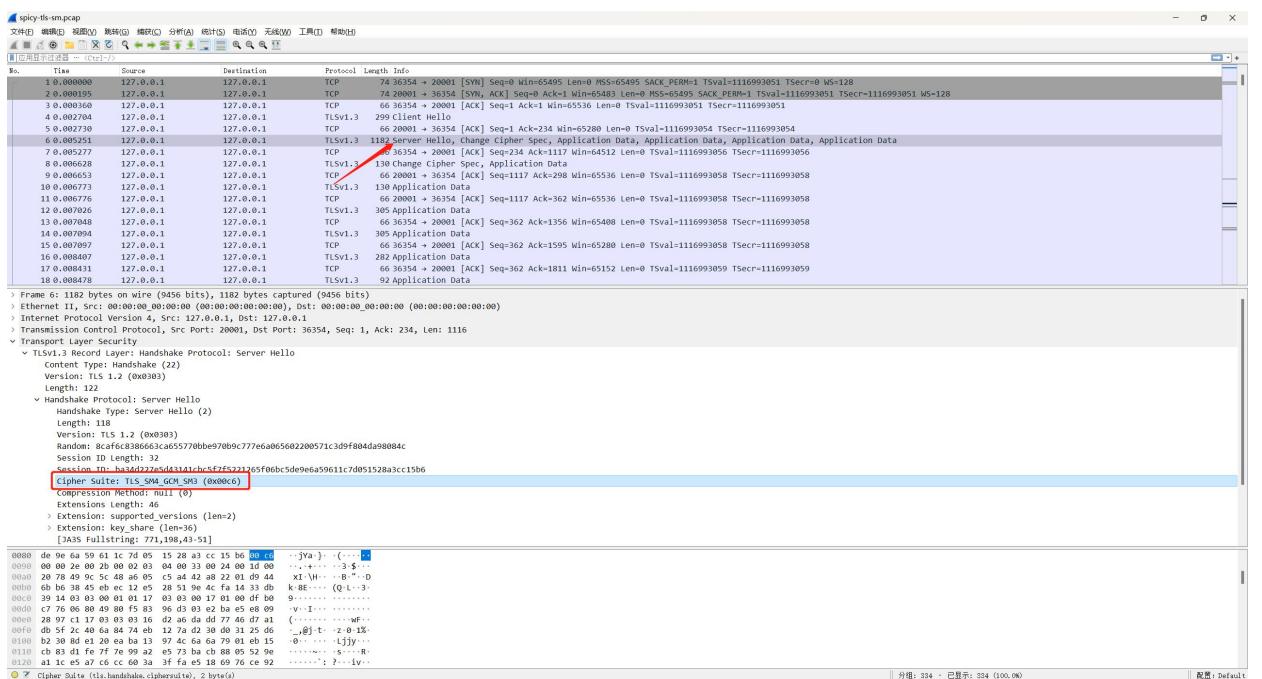
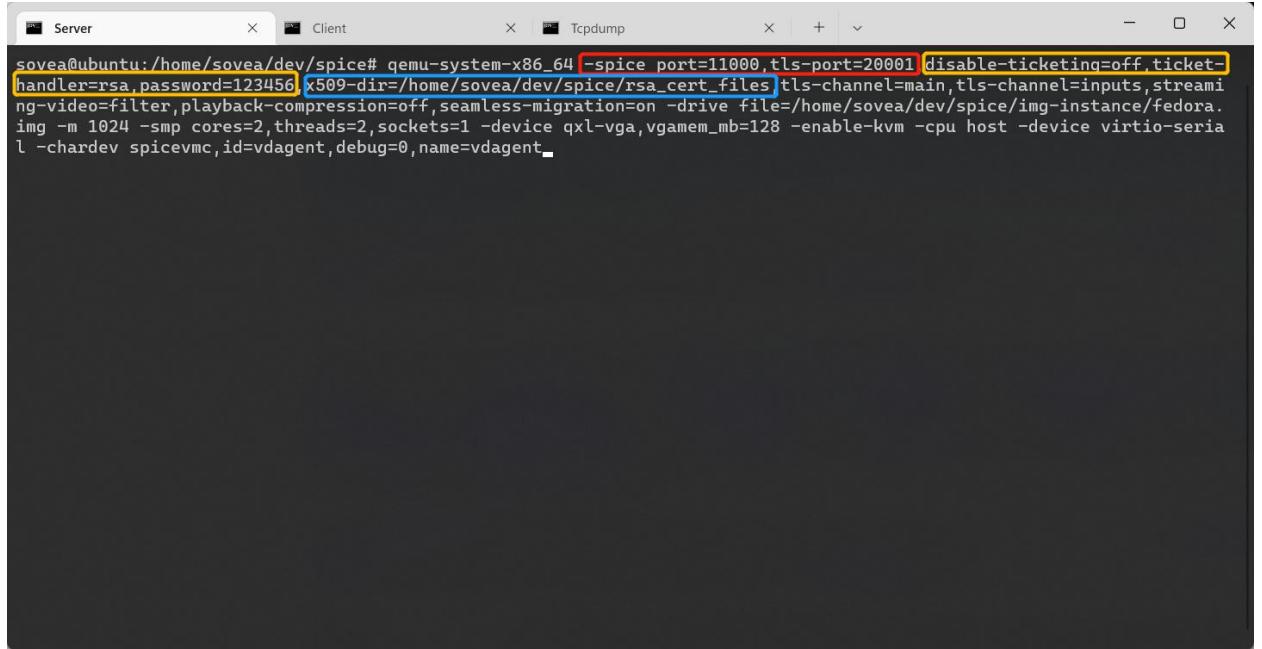


图 6 Wireshark 分析 C/S 国密连接数据包

基于非国密的 TLS 传输及 RSA 口令身份验证兼容

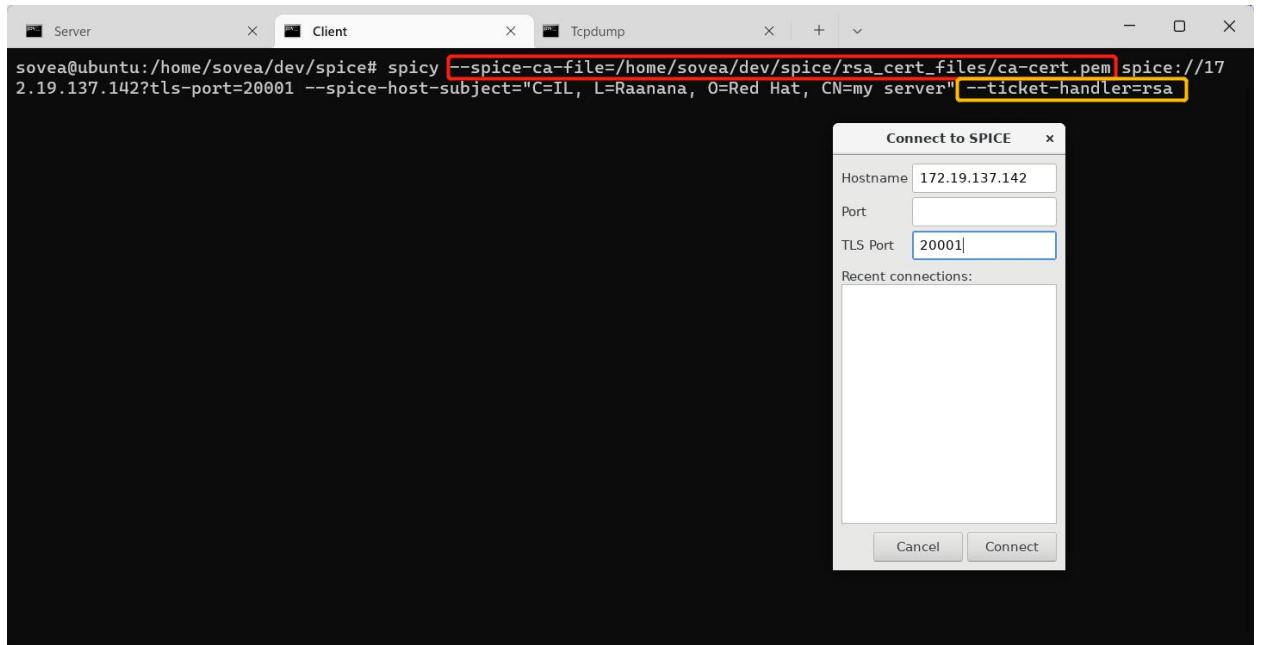
利用 qemu-system-x86_64 启动虚拟机并开启 Spice 传输，利用 RSA 证书进行加密传输，并指定 ticket-handler 参数为 rsa 在口令验证时使用 RSA 算法，x509-dir 指定 RSA 证书所在目录，其余参数设置与国密传输时保存一致，如图 7 所示。



```
sovea@ubuntu:/home/sovea/dev/spice# qemu-system-x86_64 -spice port=11000,tls-port=20001 disable-ticketing=off,ticket-handler=rsa,password=123456 x509-dir=/home/sovea/dev/spice/rsa_cert_files tls-channel=main,tls-channel=inputs,streaming-video=filter,playback-compression=off,seamless-migration=on -drive file=/home/sovea/dev/spice/img-instance/fedora.img -m 1024 -smp cores=2,threads=2,sockets=1 -device qxl-vga,vgamem_mb=128 -enable-kvm -cpu host -device virtio-serial -chardev spicevnc,id=vdagent,debug=0,name=vdagent
```

图 7 C/S 测试 QEMU 启用 Spice 非国密传输及 RSA 口令验证

Spicy 客户端指定 CA 证书、连接地址及端口、证书主题字段进行连接，同时指定参数 ticket-handler 为 rsa，使用 RSA 算法进行口令验证。输入 password 通过验证完成连接，显示远程虚拟机桌面。过程如下图 8(a) - 8(c)所示。



(a)

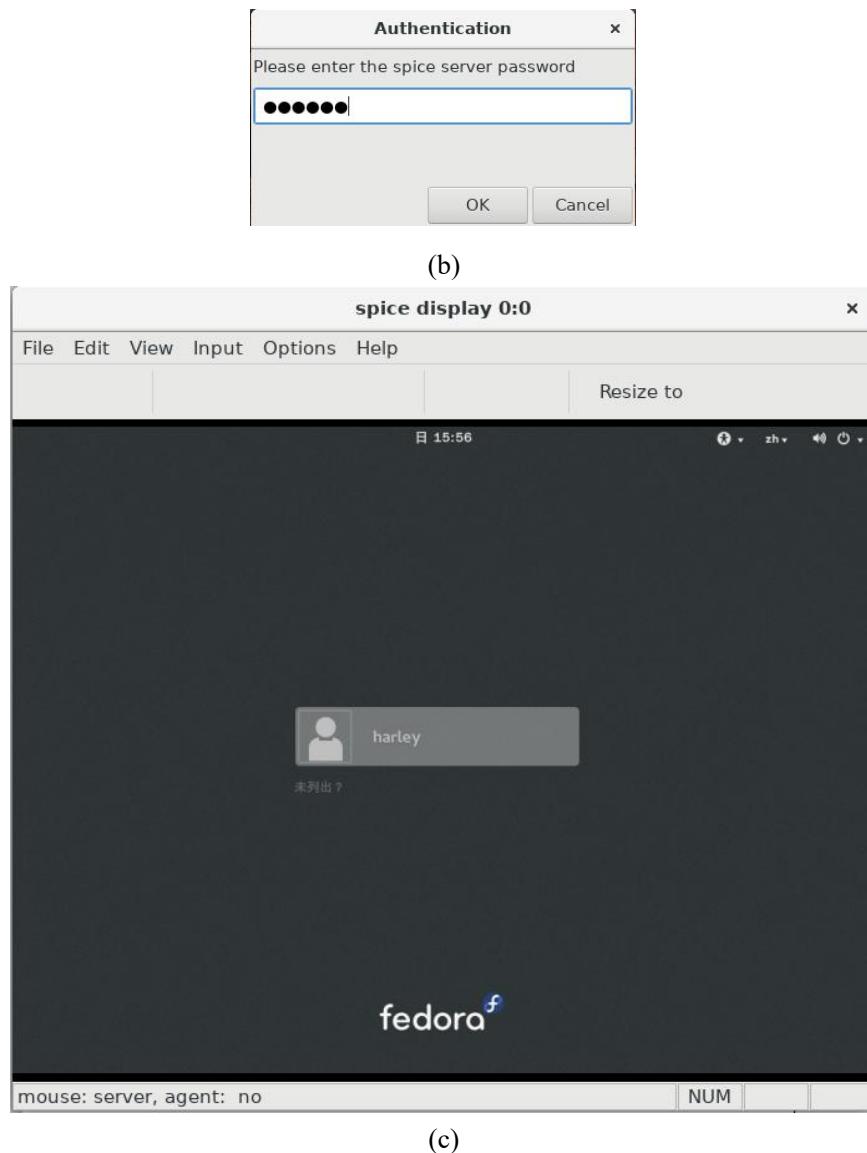


图 8 Spicy 使用 RSA 证书发起连接及 RSA 口令验证

同样使用 tcpdump 及 Wireshark 进行抓包分析，如下图 9、10 所示。根据 Wireshark 分析可知连接使用 Cipher Suite 为 TLS_AES_256_GCM_SHA384，证明当前 C/S 国密改造方案能保持原基于 RSA 证书的加密传输方案兼容。

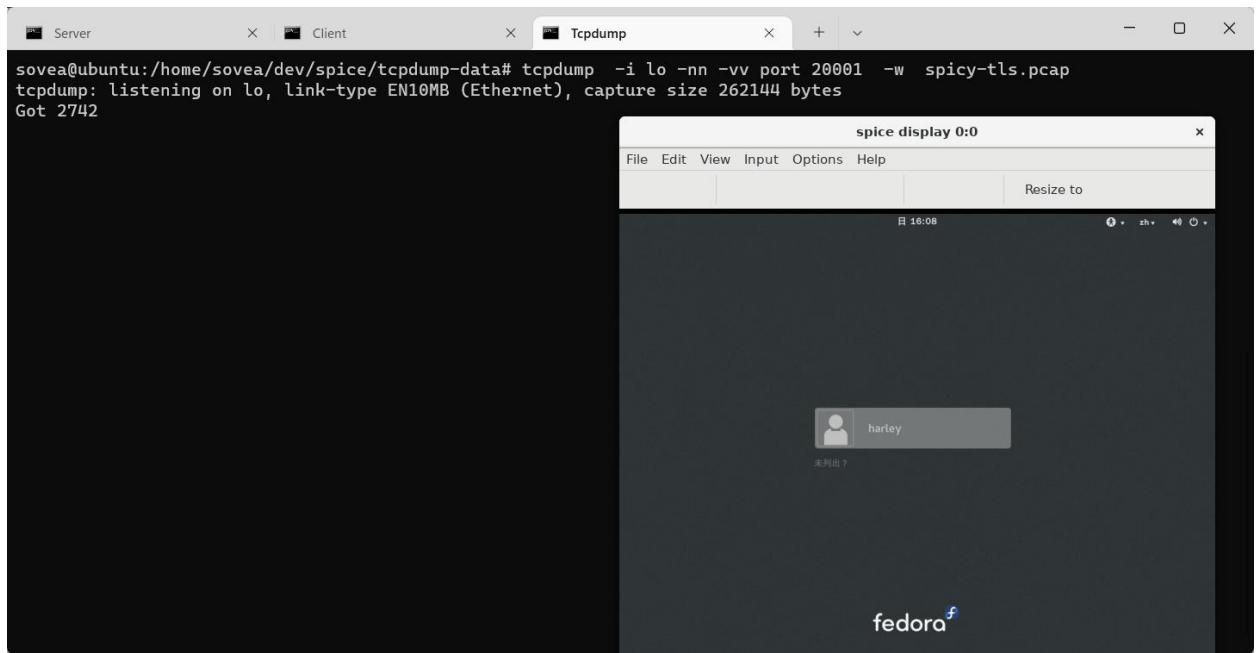


图 9 tcpdump 抓取 C/S 非国密连接数据包

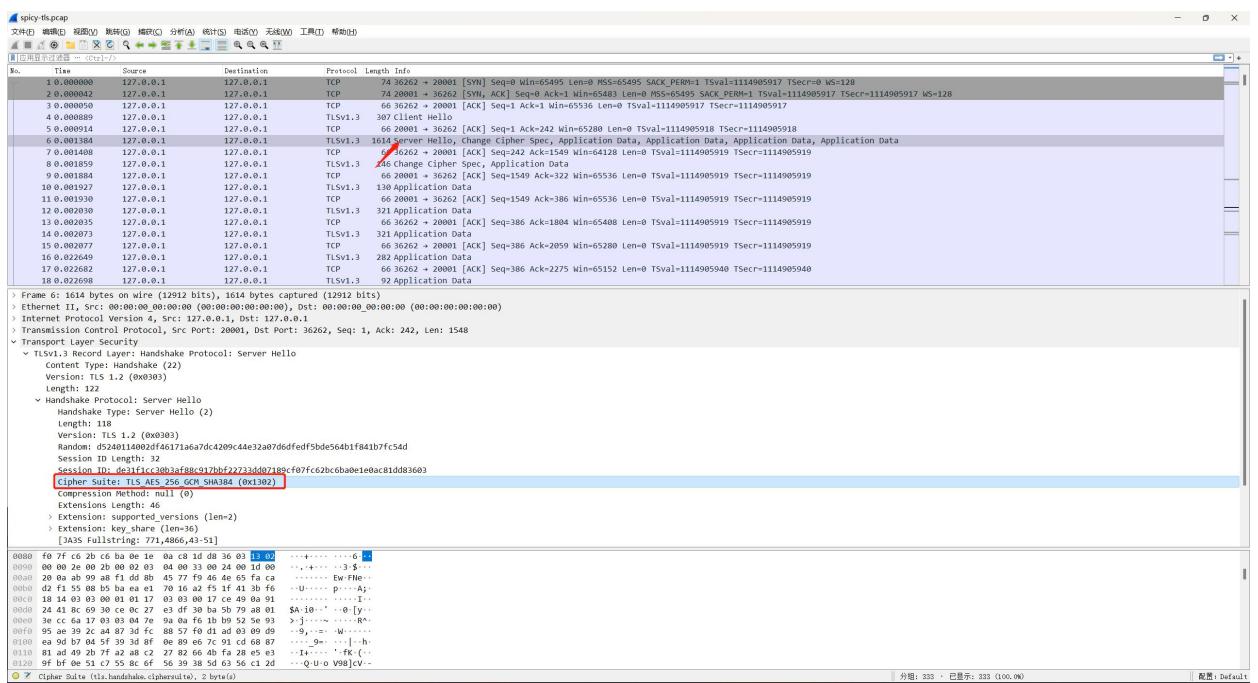


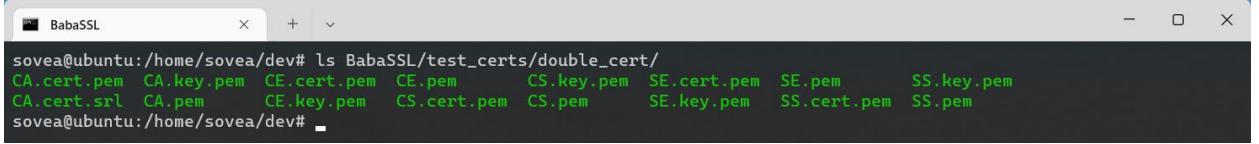
图 10 Wireshark 分析 C/S 非国密连接数据包

B/S 测试

B/S 架构测试主要利用 Server 开启国密/非国密传输以及切换口令验证使用 SM2/RSA 算法，利用 Spice-html5 连接测试，以此证明 B/S 架构下改造方案可行有效，并保持双方案兼容共存。过程中将使用兼容/不兼容国密的浏览器进行测试，以此证明方案兼容性、普适性。

证书签发及部署

首先签发符合 NTLS 的 SM2 双证书及 RSA 证书，前者可使用 BabaSSL 手册提供的 mkcert.sh 脚本或 test_certs/double_cert 目录下示例证书，如下图 11 所示。



```
sovea@ubuntu:/home/sovea/dev# ls BabaSSL/test_certs/double_cert/
CA.cert.pem  CA.key.pem  CE.cert.pem  CE.key.pem  CS.cert.pem  CS.key.pem  SE.cert.pem  SE.key.pem
CA.cert.srl  CA.pem     CE.key.pem   CS.cert.pem  CS.pem      SE.key.pem   SS.cert.pem  SS.pem
sovea@ubuntu:/home/sovea/dev#
```

图 11 SM2 双证书目录

将 Spice-html5 利用 Tengine 部署访问。改造后 Tengine 支持利用上述两套证书实现 HTTPS 兼容访问，这是为解决浏览器兼容问题。当支持国密的浏览器访问时优先使用 SM2 双证书建立连接，其余情况自动切换使用 RSA 证书保证兼容可用。具体部署方式如下图 12 所示。

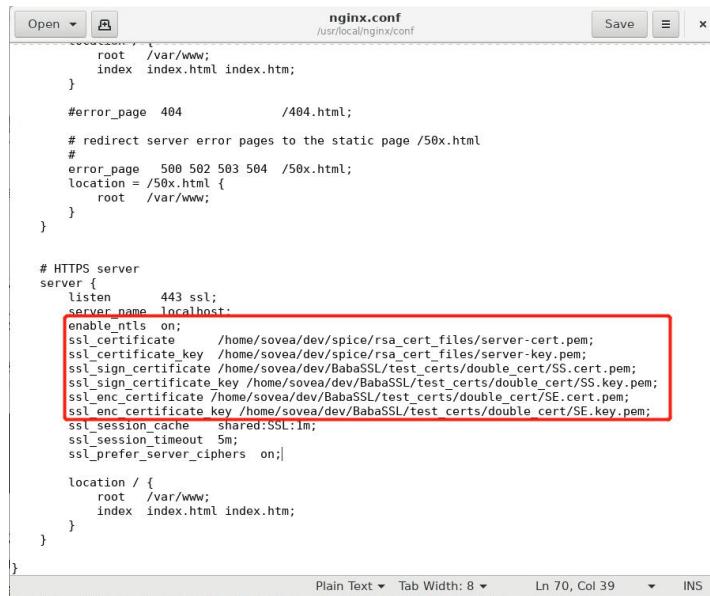


图 12 Tengine 双证书配置

测试准备

改造后 Stunnel 支持国密套件及证书，连接时加密代理方式将依据 Server 进行选择。因此，需对国密/非国密两种方案进行代理配置。当然，该方案同样适于 HTTP 访问，但测试样例采用 HTTPS 进行连接。当前以 C/S 测试接断使用的固定证书为例，Stunnel 设置相应 conf 配置文件，对国密方案指定相应套件进行加密代理。简要配置文件如下图 13(a) - 13(b) 所示。

```

securityLevel = 0
[http]
sslVersion = TLSv1.3
options = CIPHER_SERVER_PREFERENCE
client = yes
accept = 20002
connect = 127.0.0.1:20001
verify = 1
cipherSuites = TLS SM4 GCM SM3;TLS SM4 CCM SM3
cert = /home/sovea/dev/spice/sm2_cert_files/server-cert.pem
key = /home/sovea/dev/spice/sm2_cert_files/server-key.pem
CAfile = /home/sovea/dev/spice/sm2_cert_files/ca-cert.pem

```

(a)

```

securityLevel = 0
[http]
sslVersion = TLSv1.3
options = CIPHER_SERVER_PREFERENCE
client = yes
accept = 20003
connect = 127.0.0.1:20001
verify = 1
cert = /home/sovea/dev/spice/rsa_cert_files/server-cert.pem
key = /home/sovea/dev/spice/rsa_cert_files/server-key.pem
CAfile = /home/sovea/dev/spice/rsa_cert_files/ca-cert.pem

```

(b)

图 13 Stunnel 国密/非国密指定证书加密代理配置

Spice-html5 连接默认使用 WebSocket 进行连接, 利用 Tengine 实现 WSS(WebSocket Secure) 反向代理使得连接更加安全, 并可通过配置 proxy_pass 将流量转发给 Stunnel, 最终实现与 Server 的安全连接, 如下图 14 所示。

```

map $http_upgrade $connection_upgrade {
    default upgrade;
    '' close;
}

server {
    listen 8888 ssl;
    server_name localhost;

    # charset koi8-r;
    # access_log logs/host.access.log main;
    enable_ntls on;
    ssl_certificate      /home/sovea/dev/spice/rsa_cert_files/server-cert.pem;
    ssl_certificate_key  /home/sovea/dev/spice/rsa_cert_files/server-key.pem;
    ssl_sign_certificate /home/sovea/dev/BabaSSL/test_certs/double_cert/SS.cert.pem;
    ssl_sign_certificate_key /home/sovea/dev/BabaSSL/test_certs/double_cert/SS.key.pem;
    ssl_enc_certificate /home/sovea/dev/BabaSSL/test_certs/double_cert/SE.cert.pem;
    ssl_enc_certificate_key /home/sovea/dev/BabaSSL/test_certs/double_cert/SE.key.pem;
    ssl_session_timeout 5m;
    ssl_prefer_server_ciphers on;
}

location /ntls {
    proxy_ssl_session_reuse on;
    proxy_http_version 1.1;
    proxy_pass http://127.0.0.1:20002;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "Upgrade";
}

location /tls {
    proxy_ssl_session_reuse on;
    proxy_http_version 1.1;
    proxy_pass http://127.0.0.1:20003;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "Upgrade";
}

# redirect server error pages to the static page /50x.html
# error_page 500 502 503 504 /50x.html;
# location = /50x.html {
#     root /usr/service/web/error;
# }

```

图 14 Tengine WSS 代理转发

利用 Node 服务提供 SM2 插件能力时，同样利用 Tengine 配置双证书提供 HTTPS，具体配置如图 15 所示。

```
Open sm2-crypto-plugin-spice.conf /usr/local/nginx/conf/conf.d Save x
map $http_upgrade $connection_upgrade {
    default upgrade;
    '' close;
}

server {
    listen 18887 ssl;
    server_name localhost;

    # charset koi8-r;
    # access log logs/host.access.log main;
    enable_ntls on;
    ssl_certificate      /home/sovea/dev/spice/rsa_cert_files/server-cert.pem;
    ssl_certificate_key  /home/sovea/dev/spice/rsa_cert_files/server-key.pem;
    ssl_sign_certificate /home/sovea/dev/BabaSSL/test_certs/double_cert/SS.cert.pem;
    ssl_sign_certificate_key /home/sovea/dev/BabaSSL/test_certs/double_cert/SS.key.pem;
    ssl_enc_certificate /home/sovea/dev/BabaSSL/test_certs/double_cert/SE.cert.pem;
    ssl_enc_certificate_key /home/sovea/dev/BabaSSL/test_certs/double_cert/SE.key.pem;
    ssl_session_timeout 5m;
    ssl_prefer_server_ciphers on;
    location / {
        proxy_ssl_session_reuse on;
        proxy_http_version 1.1;
        proxy_pass http://127.0.0.1:18888;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "Upgrade";
    }

    # redirect server error pages to the static page /50x.html
    # error_page 500 502 503 504 /50x.html;
    # location = /50x.html {
    #     root /usr/service/web/error;
    # }
}
```

Plain Text Tab Width: 8 ▾ Ln 35, Col 2 ▾ INS

图 15 Node SM2 插件 Tengine 配置

基于国密/非国密的浏览器 HTTPS 访问

利用支持国密套件的可信浏览器、360 浏览器国密版查看自签发证书详情如下图 16 所示，可知当前 HTTPS 连接利用 SM2 国密双证书建立，并使用 SM4_GCM 进行加密和身份验证，SM2 作为密钥交换机制。

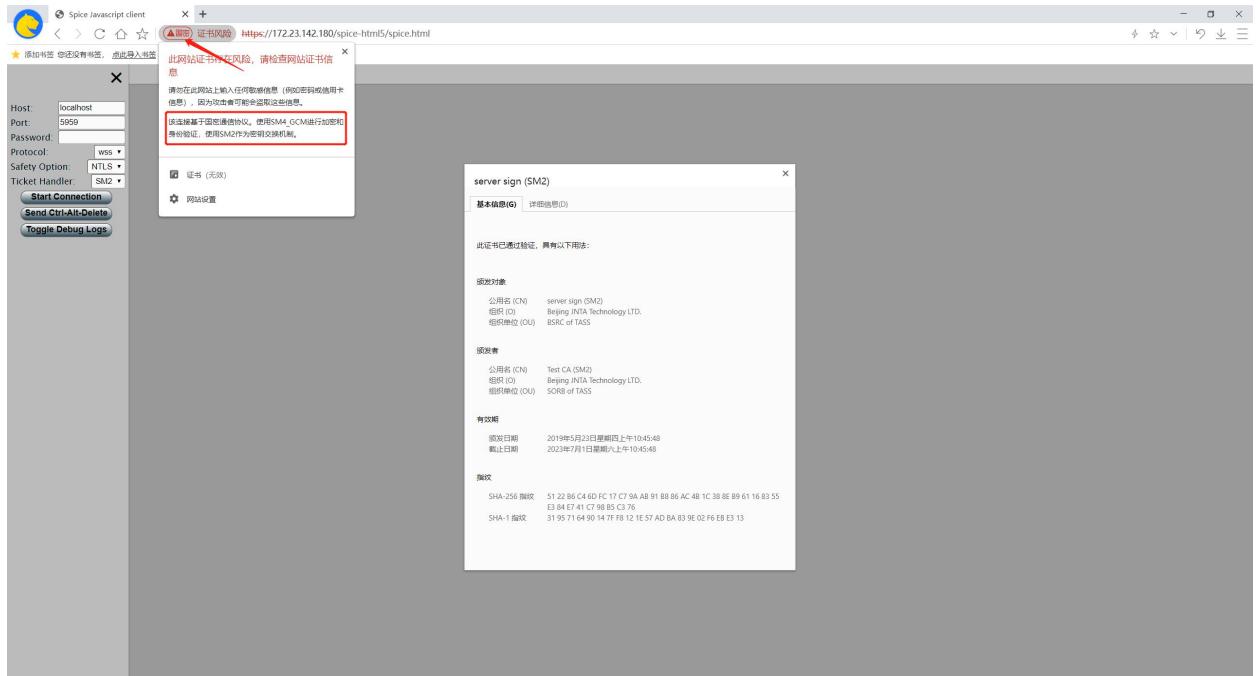


图 16 兼容国密套件浏览器访问详情

使用不兼容国密套件的 Chrome 等访问，自签发证书详情如图 17 所示，可知当前 HTTPS 连接建立使用 RSA 证书保持兼容访问。

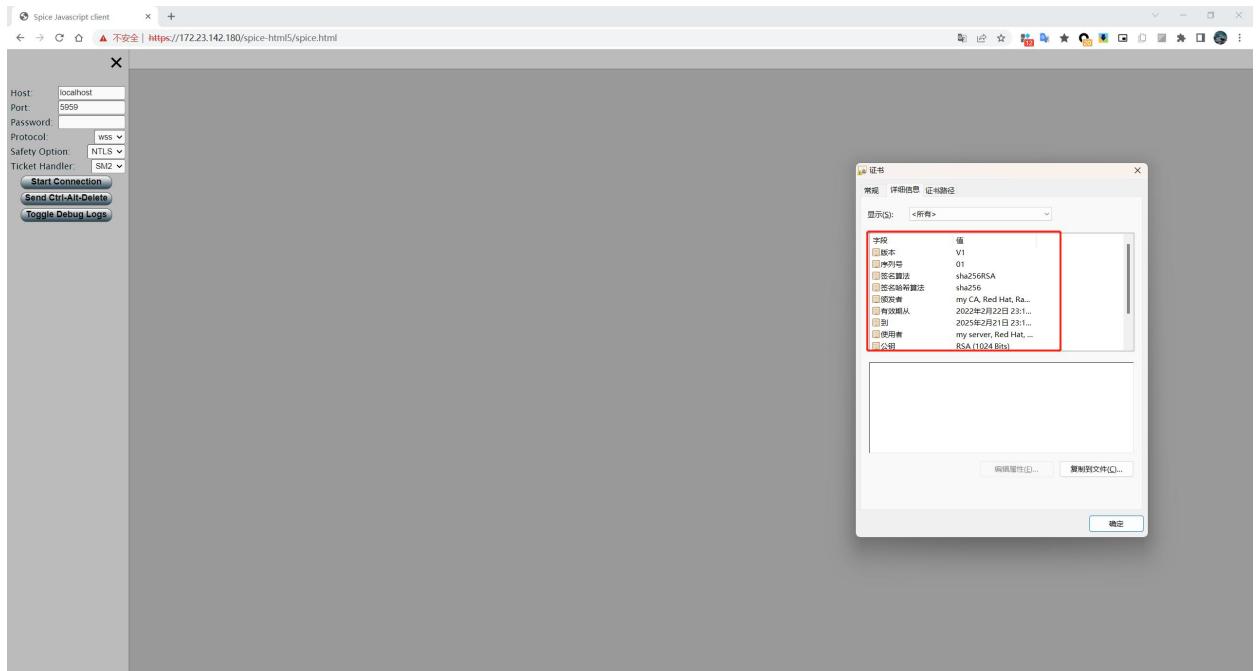
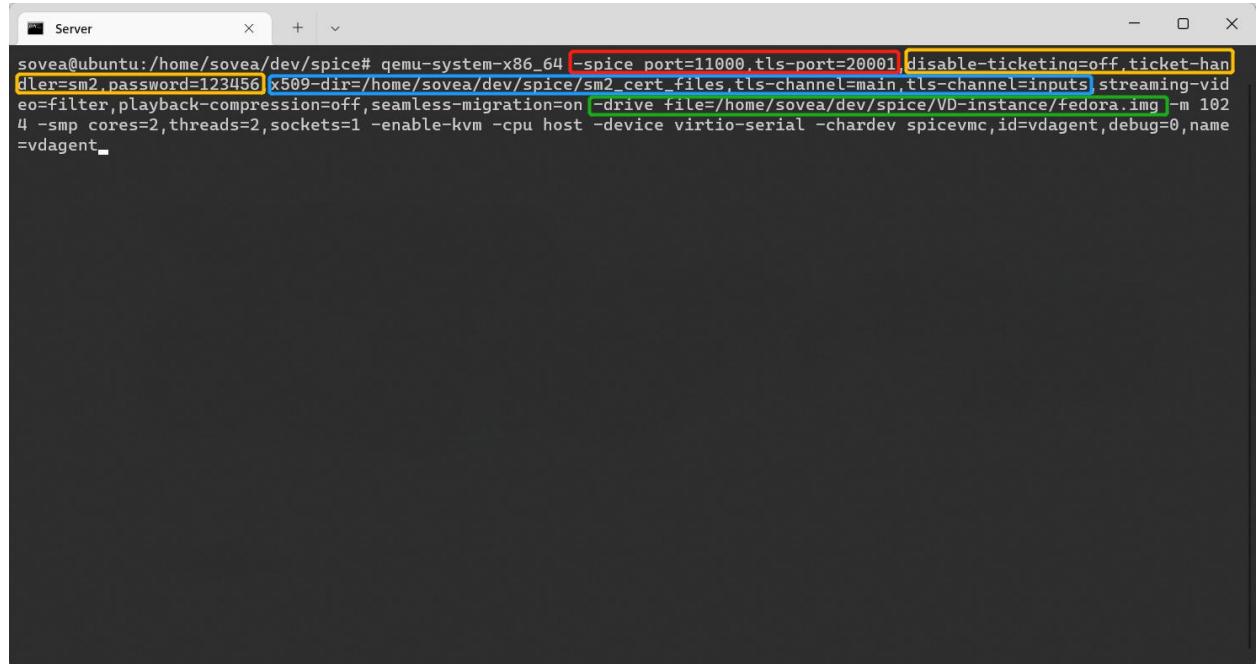


图 17 不兼容国密套件浏览器访问详情

基于国密的 TLS 传输及 SM2 口令身份验证

为证明连接过程中浏览器兼容性对加密方式选择无影响的特性，主要采用 Chrome 进行测试。利用 qemu-system-x86_64 启动虚拟机并开启 Spice 传输，利用 SM2 单证书进行国密

传输，并指定 ticket-handler 参数为 sm2(默认)在口令验证时使用 SM2 算法。设置默认端口及 tls 端口，开启口令验证，设置 password，x509-dir 指定 SM2 单证书所在目录，Spice 将据此读取所需 CA、Server 证书，tls-channel 指定需加密通道，运行前文所述安装的 Fedora 虚拟机，如图 18 所示。



```
sovea@ubuntu:/home/sovea/dev/spice# qemu-system-x86_64 -spice_port=11000,tls-port=20001,disable-ticketing=off,ticket-handlers=sm2,password=123456,x509-dir=/home/sovea/dev/spice/sm2_cert_files,tls-channel=main,tls-channel=inputs,streaming-video-filter,playback-compression=off,seamless-migration=on -drive file=/home/sovea/dev/spice/VD-instance/fedora.img -m 1024 -smp cores=2,threads=2,sockets=1 -enable-kvm -cpu host -device virtio-serial -chardev spicevnc,id=vdaagent,debug=0,name=vdaagent
```

图 18 B/S 测试 QEMU 启用 Spice 国密传输及 SM2 口令验证

Spice-html5 检测当前 HTTPS 访问，切换到 WSS 协议，设定端口为 Tengine 代理 8888 端口，通过 Server 参数配置选择 NTLS 加密方式及 SM2 算法处理口令验证，点击 Start Connection 发起连接，如下图 19 所示。

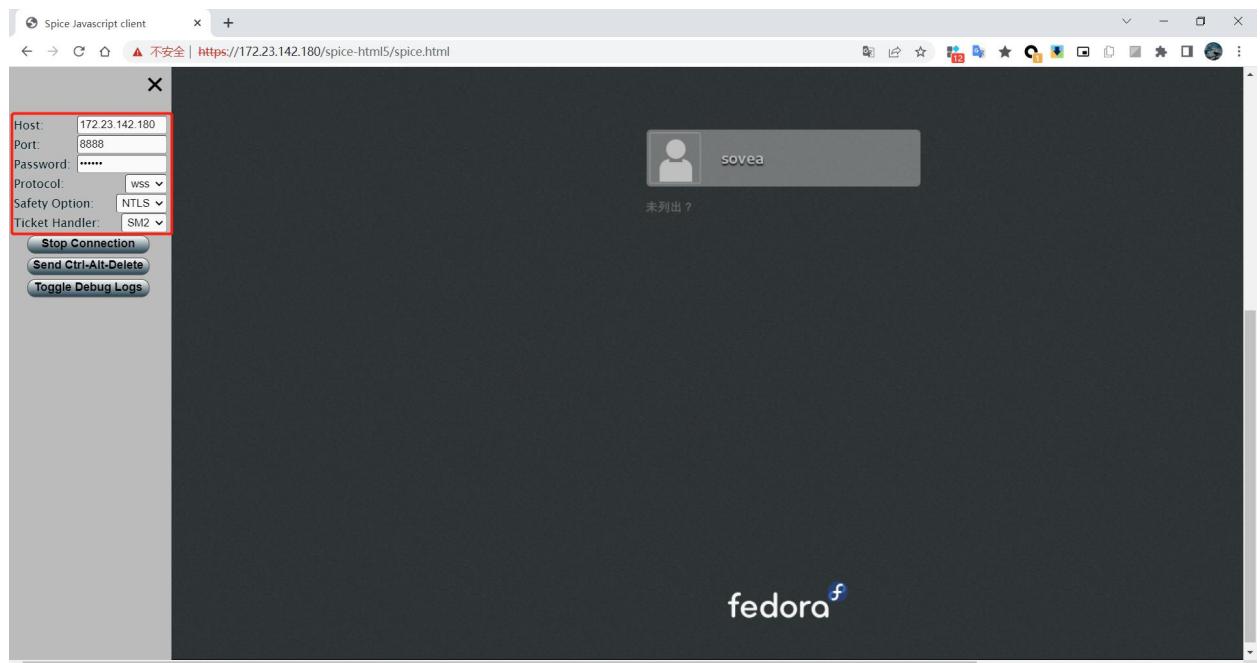


图 19 Spice-html5 连接国密传输及 SM2 口令验证 Server

使用 tcpdump 对 Server 监听 tls-port 进行抓包，利用 Wireshark 进行分析可查看当前 TLS 传输连接的密钥协商结果，从而验证 B/S 架构下国密改造完成情况，如下图 20、21 所示。

根据 Wireshark 分析可知连接使用 Cipher Suite 为 TLS_SM4_GCM_SM3，证明当前 B/S 国密改造方案有效。

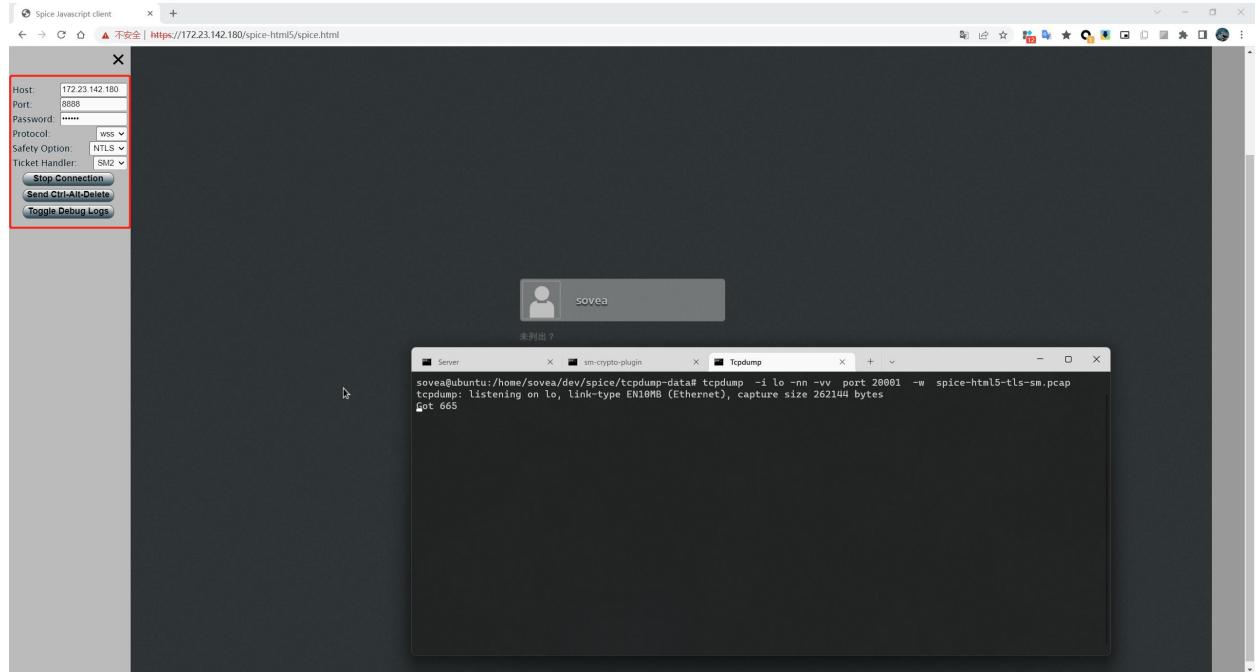


图 20 tcpdump 抓取 B/S 国密连接数据包

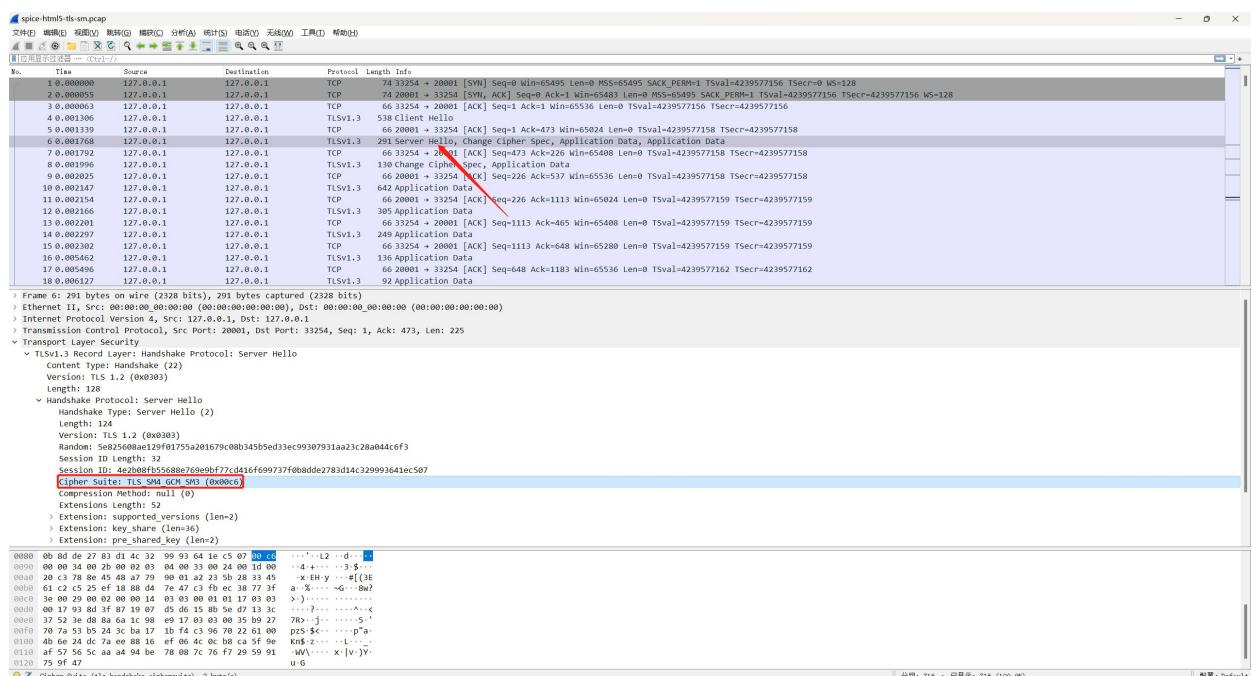
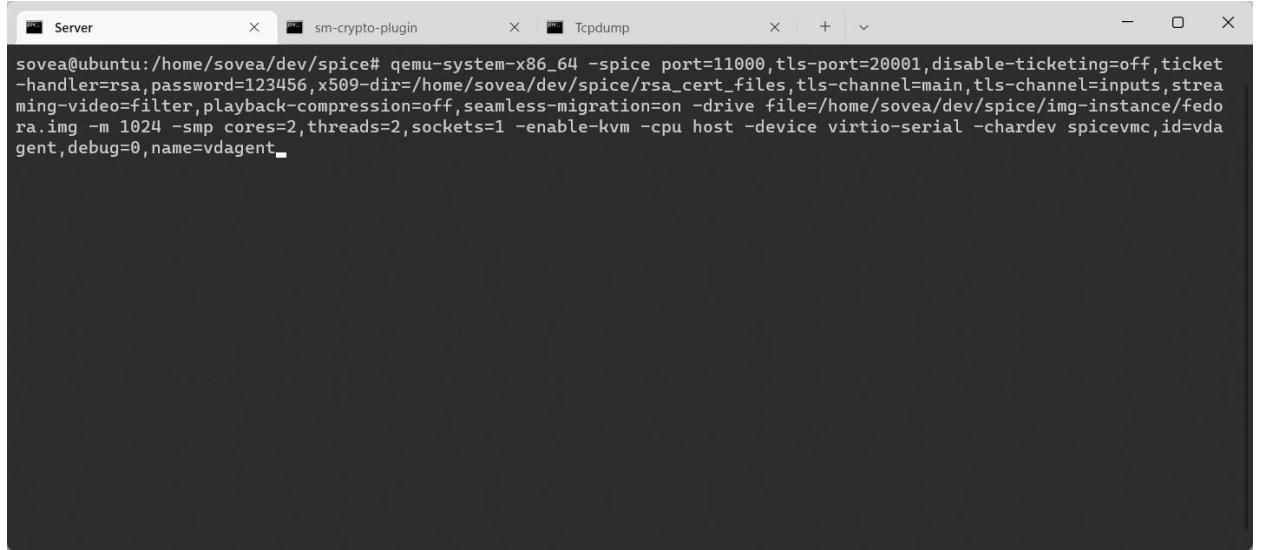


图 21 Wireshark 分析 B/S 国密连接数据包

基于非国密的 TLS 传输及 RSA 口令身份验证兼容

利用 qemu-system-x86_64 启动虚拟机并开启 Spice 传输，利用 RSA 证书进行加密传输，并指定 ticket-handler 参数为 rsa 在口令验证时使用 RSA 算法，x509-dir 指定 RSA 证书所在目录，其余参数设置与国密传输时保存一致，如图 22 所示。



```
sovea@ubuntu:/home/sovea/dev/spice# qemu-system-x86_64 -spice port=11000,tls-port=20001,disable-ticketing=off,ticket-handler=rsa,password=123456,x509-dir=/home/sovea/dev/spice/rsa_cert_files,tls-channel=main,tls-channel=inputs,stream-video=filter,playback-compression=off,seamless-migration=on -drive file=/home/sovea/dev/spice/img-instance/fedora.img -m 1024 -smp cores=2,threads=2,sockets=1 -enable-kvm -cpu host -device virtio-serial -chardev spicevmc,id=vda,gent,debug=0,name=vdagent
```

图 22 B/S 测试 QEMU 启用 Spice 非国密传输及 RSA 口令验证

Spice-html5 检测当前 HTTPS 访问，切换到 WSS 协议，设定端口为 Tengine 代理 8888 端口，通过 Server 参数配置选择 TLS 加密方式及 RSA 算法处理口令验证，点击 Start Connection 发起连接，如下图 23 所示。

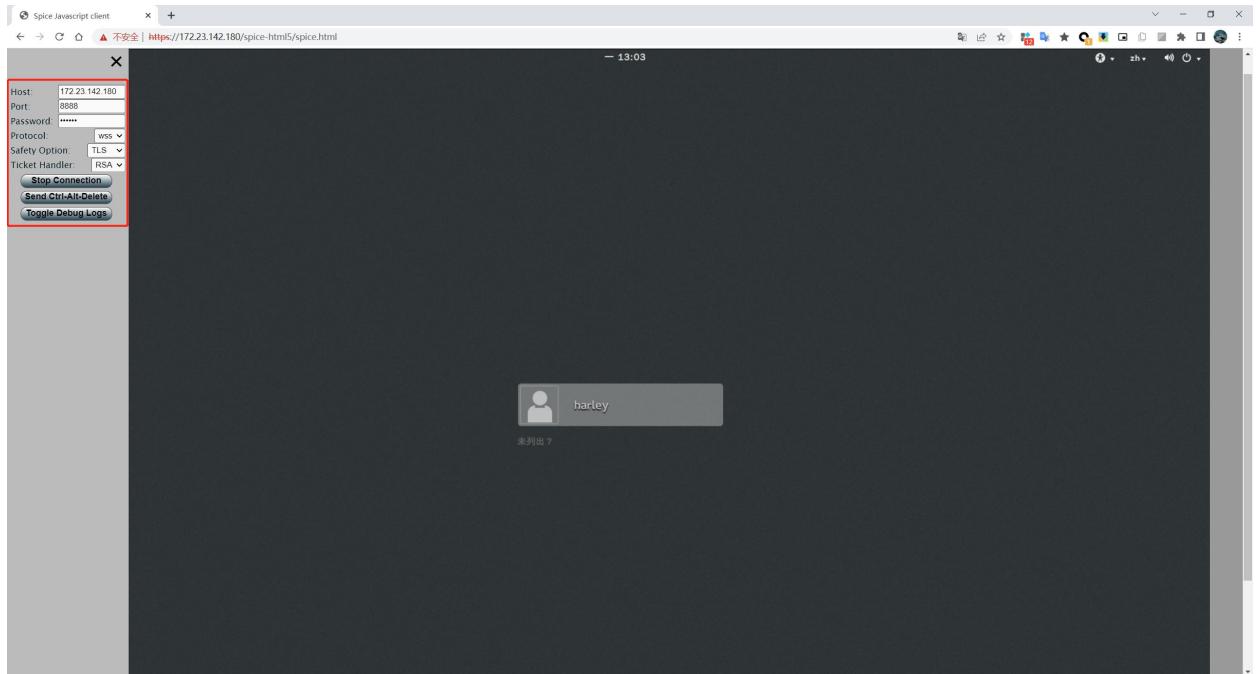


图 23 Spice-html5 连接非国密传输及 RSA 口令验证 Server

使用 tcpdump 对 Server 监听 tls-port 进行抓包，利用 Wireshark 进行分析可查看当前 TLS

传输连接的密钥协商结果，从而验证 B/S 架构下非国密方案兼容情况，如下图 24、25 所示。

根据 Wireshark 分析可知连接使用 Cipher Suite 为 TLS_AES_256_GCM_SHA384，证明当前 B/S 国密改造方案能保持基于 RSA 证书的加密传输方案兼容。

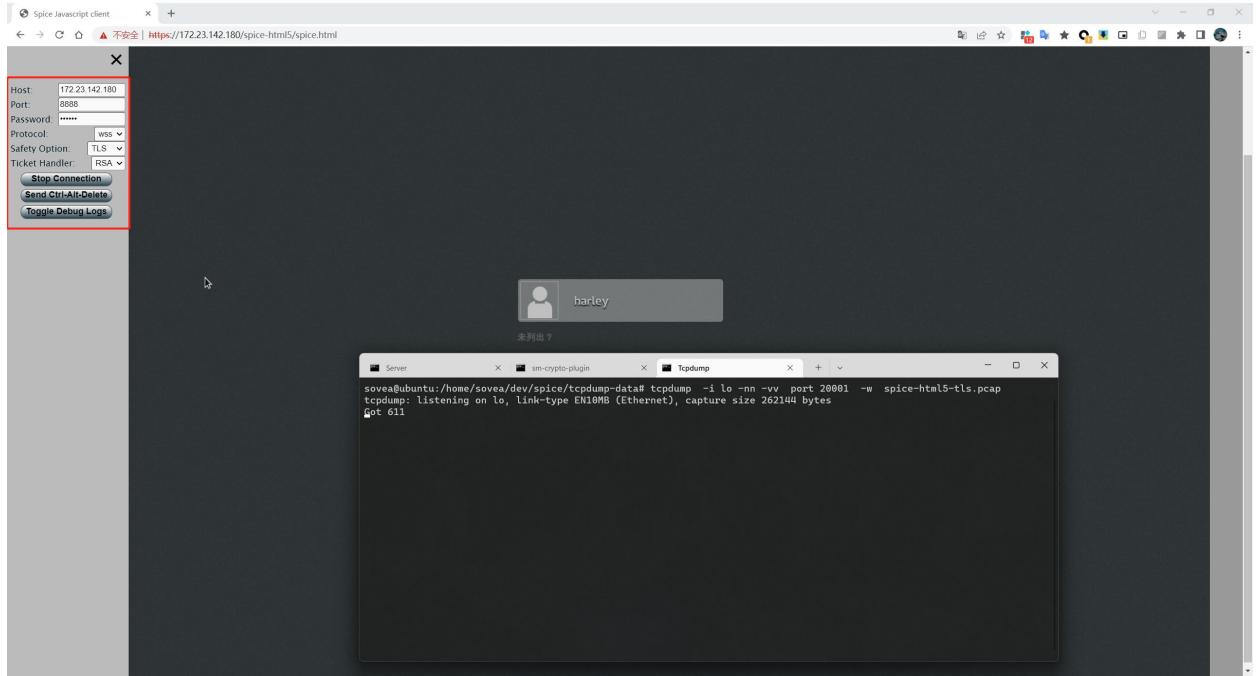


图 24 tcpdump 抓取 B/S 非国密连接数据包

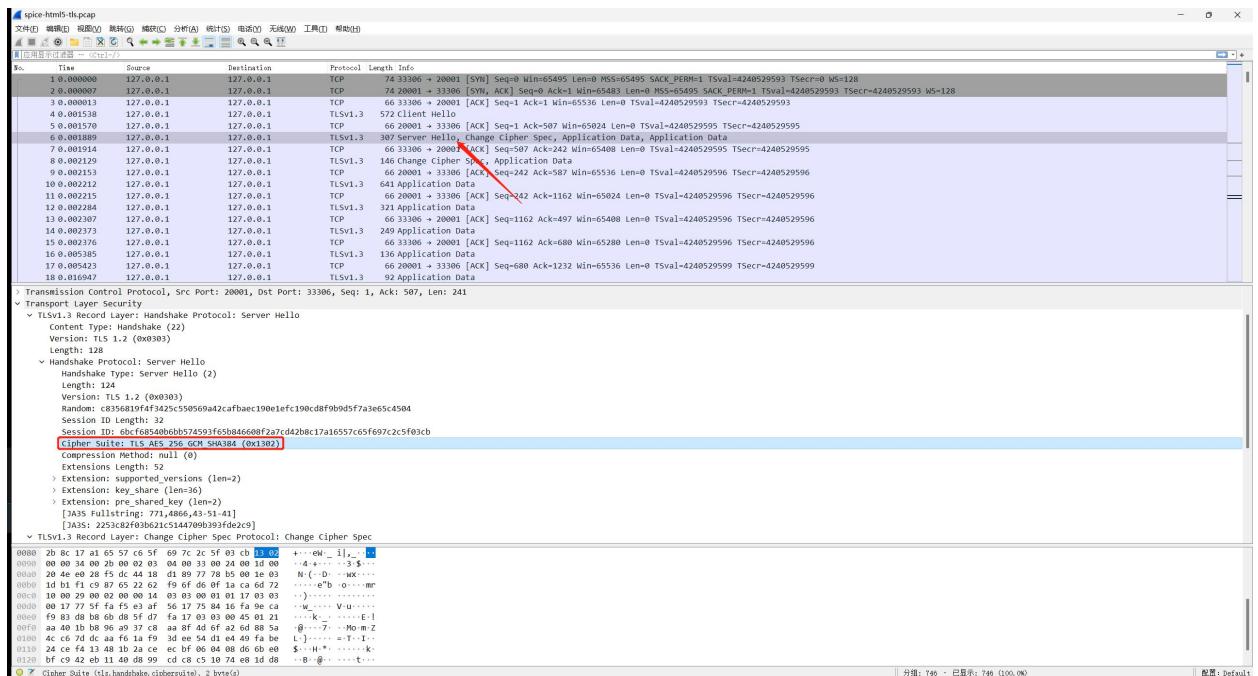


图 25 Wireshark 分析 B/S 非国密连接数据包