AUTHENTICATION AND ENCRYPTION HOWTO

Due to the private nature of patient data and proprietary PK models, authentication and encryption are critical to the functioning of Spk. This document provides specific directions for installing the required software infrastructure for the RFPK Spk demo system.

Table of Contents

Introduction

The authentication and encryption infrastructure must achieve the following goals:

1. A user logging into the web site must have confidence that this is, indeed, the proper site and not an impostor.
2. All communication between Spk components that travels over the public Internet must be encrypted.
3. The web server must have confidence that the client application with which it is communicating is a valid unmodified version of software released by RFPK.

The first two goals are accomplished by configuring the web server to use the Secure Socket Layer (SSL) and by installing a validated authentication certificate.

Authentication is performed whenever a user selects a URL that starts with "https://", indicating that the secure version of the http protocol is to used, the version that incorporates the Secure Socket Layer (SSL). In the process of establishing an SSL connection to the web server, the user's browser is sent a copy of the certificate by the server. To be accepted, the certificate must have been digitally signed by a Certificate Authority (CA) whose signature and public encryption key are already on file within the browser. Using this encryption key, the browser deciphers the signature field in the certificate, and if the answer is indeed the expected signature of the CA, the connection process goes forward.

Over an SSL connection, every message is encrypted, in either direction. A 128-bit cipher is used, providing the measure of security that is now standard for the best commercial web sites.

The third goal is accomplished by the Java WebStart technology used to distribute new and updated versions the Spk client, called the MDA. The MDA itself is digitally signed by the RFPK software team. WebStart is configured so that only properly signed software will run.

Good Housekeeping

We will start by creating a file to hold our certificates, certificate request, and our certificate password. Not everyone will agree that it is a good idea to keep a plain text version of the password in a file, but here is the reasoning:

1. The certificate is worthless if the password is forgotten.
2. The password must be provided to the tomcat webserver as an argument, each time it starts up. In order for tomcat to start automatically when the computer reboots, it is necessary to place the password, in plain text, in a configuration file called `server.xml`. Unless the server is always started manually, there is no way to avoid storing the password, somewhere, in plain text.
3. Since we must store the password in plain text, the important thing is to be careful with file access permissions on the files in which we store it.

Let's get started, then:

```
su -
mkdir cert
chmod 750 cert
cd cert
```

Next think up a very good password. Store a copy of it in a file called password, so that it will not be forgotten:

```
echo 'Make@!This&*Good' > password
chmod 440 password
```

Obtaining a Certificate

A certificate must be signed by a Certificate Authority (CA). The best known CAs, including Verisign, Thawte, GeoTrust, provide this service for a fee. They are respected and have been in the business long enough that most browsers come with copies of their root certificates pre-installed. Any certificate sold by one of these well-known CAs will be derivative of one or another of these pre-installed root certificates and, hence, will be accepted automatically by 99% of the browsers, without requiring any user feedback.

The University of Washington (UW) is a CA. In fact, any one can self-sign a certificate and hence act as a CA. A certificate signed by the University of Washington is apt to accepted by a broader spectrum of users than one signed by Alan Westhagen, Jiaji Du, or even RFPK. The UW root certificate is not, however, pre-installed in browsers. To access secure services at the UW, users are asked to install the UW Services Certificate [1], which can done automatically, with a button click, with most browsers.

The following sections will describe the process for obtaining a certificate from the University of Washington CA. The first step will be to generate a public/private key pair. Once you have these keys, you will generate a properly formatted Certificate Signing Request which will be needed by the CA. You will then be ready to use the semi-automated process on a UW Computing and Communications for placing a request to have a certificate generated. Finally you will, if all goes well, retrieve the certificate.

Generate a Public and a Private Key

The first step is to produce a randomly generated public/private encryption key pair. The keys are placed in a binary file called `.keystore` in the caller, i.e. root, home directory by default, which is the place we want to keep it in. Tomcat servers require RSA encyription algorithm and look for an entry in `.keystore` named `tomcat`.

```
su -
$JAVA_HOME/bin/keytool -genkey -alias tomcat -keyalg RSA
```

You will be asked to provide a password. Make it a good one, because this password will provide access to the key pair that you are about to generate.

The **keytool** program will ask you a number of questions. It is most important that you provide a good password when requested, because this will protect secrecy of your public/private key pair. Be certain, as well, to provide the full domain name of the server, rather than your own first and last name.

Here is the list of questions and suggested responses. The questions are in *italics* and the responses in normal text. In response to the last question, simply press **Enter**.

```
Enter keystore password: Make@!This&*Good

What is your first and last name?
    [Unknown]: spk.rfpk.washington.edu

What is the name of your organizational unit?
    [Unknown]: RFPK

What is the name of your organization?
    [Unknown]: University of Washington

What is the name of your City or Locality?
    [Unknown]: Seattle

What is the name of your State or Province?
    [Unknown]: WA

What is the two-letter country code for this unit?
    [Unknown]: US

Is CN=spk.rfpk.washington.edu, OU=University of Washington, O=RFPK, L=Seattle, ST=WA, C
    [no]: yes

Enter key password for <tomcat>
    (RETURN if same as keystore password):
```

Generate a Certificate Signing Request

Now that you have your keys, you can generate a CSR.

```
cd cert
$JAVA_HOME/bin/keytool -certreq -keyalg RSA -alias tomcat -file certreq.csr
```

The resulting text file, `certreq.csr`, will look something like this:

```
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIBwTCCASoCAQAwgYAxCzAJBgNVBAYTAlVTMQswCQYDVQQIEwJXQTEQMA4GA1UEBxMHU2VhdHRs
ZTEhMB8GA1UEChMYVW5pdmVyc2l0eSBvZiBXYXNoaW5ndG9uMQ0wCwYDVQQLEwRSRlBLMSAwHgYD
VQQDExdzcGsucmZway53YXNoaW5ndG9uLmVkdTCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEA
vwQZDAkPEmB4H3QKM3mVC6CsgD9a2jBB4DXT5olQ1/t0nCwZFzqkGNhwEZ8/gIRsjDRYgoaF+ZsD
nHfcdWDh4owS8vWTMgrDDGQDimxs8SA00aUSbGPQnzuCD9Mbjp9xpdVYtAxFU9MFRbgV1q/Atmye
i3DafG51UevT787g9QsCAwEAAaAAMA0GCSqGSIb3DQEBBAUAA4GBAElfNnSOBfI1F80XvlYu+V5f
8hTYDbKk+2rU+avZNAIHpqWezypbw/Ozi1xUwA5BNzSjzNPV/xM7BnrzcLlAuoEhVE4CJuvnJDoZ
uYfnasj1GFTNiIoKhBIyNV5GIYpB12FoQdq+erBugJU7LrEI1AGI0iBRykXzRJyLaFW9Kg7l
-----END NEW CERTIFICATE REQUEST-----
```

Request a Signed Certificate

Now go to the University of Washington web site[2] that partially automates the process of obtaining a certificate. To use it, you must have a MyNetId weblogin. You must also be listed in the DNS record as one of the system administrators for the server for which you are obtaining a certificate. If you are not sure whether or not you are in the system administrator list, start the process anyway, because checking the DNS record and adding you, if necessary, is one of the steps that will follow.

After providing information about the purpose of the certificate, you will be asked to paste a your CSR into a field in a form on the screen. In a terminal window, simply type

```
cat certreq.csr
```

to display your CSR, use your left mouse button to high-light the text from the first - sign to the last, then cut and paste the high-lighted text into the browser window. Finish the procedure at the web site, and then wait until your request is processed. If everything works correctly, you should receive email notification that your signed certificate is ready in just a few minutes.

Retrieve your Certificate

When you receive email notification that your certificate is ready, return to the University of Washington CA web site[3], select the **View requests** menu item, log in with your MyNetId weblogin if required, and click on the sequence number associated with your request. At the next page, click on **Retrieve this certificate**, and you should see something like this:

```
-----BEGIN CERTIFICATE-----
MIIECjCCA3OgAwIBAgICC5kwDQYJKoZIhvcNAQEFBQAwgZQxCzAJBgNVBAYTAlVT
MQswCQYDVQQIEwJXQTEhMB8GA1UEChMYVW5pdmVyc2l0eSBvZiBXYXNoaW5ndG9u
MRQwEgYDVQQLEwtVVyBTZXJ2aWNlczEXMBUGA1UEAxMOVVcgU2VydmljZXMgQ0Ex
JjAkBgkqhkiG9w0BCQEWF2hlbHBAY2FjLndhc2hpbmd0b24uZWR1MB4XDTA1MDEy
NTIwNDUwMFoXDTA2MDEyNTIwNDUwMFowgYAxCzAJBgNVBAYTAlVTMQswCQYDVQQI
EwJXQTEQMA4GA1UEBxMHU2VhdHRsZTEhMB8GA1UEChMYVW5pdmVyc2l0eSBvZiBX
YXNoaW5ndG9uMQ0wCwYDVQQLEwRSRlBLMSAwHgYDVQQDExdzcGsucmZway53YXNo
aW5ndG9uLmVkdTCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAvwQZDAkPEmB4
H3QKM3mVC6CsgD9a2jBB4DXT5olQ1/t0nCwZFzqkGNhwEZ8/gIRsjDRYgoaF+ZsD
nHfcdWDh4owS8vWTMgrDDGQDimxs8SA00aUSbGPQnzuCD9Mbjp9xpdVYtAxFU9MF
RbgV1q/Atmyei3DafG51UevT787g9QsCAwEAAaOCAXswggF3MAwGA1UdEwEB/wQC
MAAwHQYDVR0lBBYwFAYIKwYBBQUHAwIGCCsGAQUFBwMBMB0GA1UdDgQWBBSzGZJE
ZcvFllISikcriPH5VUmAZzAiBgNVHREEGzAZghdzcGsucmZway53YXNoaW5ndG9u
LmVkdTCBwQYDVR0jBIG5MIG2gBRV18EzxvqT+Cc9yyBL9VqOWJd9dKGBmqSBlzCB
lDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMSEwHwYDVQQKExhVbml2ZXJzaXR5
IG9mIFdhc2hpbmd0b24xFDASBgNVBAsTC1VXIFNlcnZpY2VzMRcwFQYDVQQDEw5V
VyBTZXJ2aWNlcyBDQTEmMCQGCSqGSIb3DQEJARYXaGVscEBjYWMud2FzaGluZ3Rv
```

```
bi5lZHWCAQAwQQYDVR0fBDowODA2oDSgMoYwaHR0cDovL2NlcnRzLmNhYy53YXNo
aW5ndG9uLmVkdS9VV1NlcnZpY2VzQ0EuY3JsMA0GCSqGSIb3DQEBBQUAA4GBAIHI
stvbQRhuOn+c3QqZKg/hjfEpjJgiBT7269qM3uEJ6uiRmnZRY+8uI5sZjhBjd1nN
81I//GcrOYOef2dqhIB0vpO+775GBTs/3sJ44VGERCUpqOz+4/thOlUpTAtJcpEe
eLFL6SS8Ph0PWOmwAHQmebj0BpC2zJF1dgMrUGlg
-----END CERTIFICATE-----
```

Note the text above is that of the current the spk certificate valid from 01/25/2005 to 01/25/2006.

In a terminal window, go to the the root user's home directory and start your favorite editor:

```
su -
cd cert
emacs spk_cert.pem
```

to create a file to store your certificate, and cut and paste the certificate from the browser window into the editor window, starting from the first - sign to the last. Save the file.

Getting the University of Washington Root Certificate

The spk certificate that you just obtained derives its authority from the root certificate of the University of Washington. The tomcat web server will need copies of both of these certificates. The UW root is available on a UW web page but, for convenience, a copy appears, below, in this document. We will cut and paste it into a file, similar to the one that we created to hold the spk certificate.

In a terminal window, in the root user's home directory, start your favorite editor:

```
su -
cd cert
emacs uw_root_cert.pem
```

to create the file to contain the certificate, then cut and paste the following, which is the text of the UW root certificate,

```
-----BEGIN CERTIFICATE-----
MIIEBzCCA3CgAwIBAgIBADANBgkqhkiG9w0BAQQFADCBlDELMAkGA1UEBhMCVVMx
CzAJBgNVBAgTAldBMSEwHwYDVQQKExhVbml2ZXJzaXR5IG9mIFdhc2hpbmd0b24x
FDASBgNVBAsTC1VXIFNlcnZpY2VzMRcwFQYDVQQDEw5VVyBTZXJ2aWNlcyBDQTEm
MCQGCSqGSIb3DQEJARYXaGVscEBjYWMud2FzaGluZ3Rvbi5lZHUwHhcNMDMwMjI1
MTgyNTA5WhcNMzAwOTAzMTgyNTA5WjCBlDELMAkGA1UEBhMCVVMxCzAJBgNVBAgT
AldBMSEwHwYDVQQKExhVbml2ZXJzaXR5IG9mIFdhc2hpbmd0b24xFDASBgNVBAsT
C1VXIFNlcnZpY2VzMRcwFQYDVQQDEw5VVyBTZXJ2aWNlcyBDQTEmMCQGCSqGSIb3
DQEJARYXaGVscEBjYWMud2FzaGluZ3Rvbi5lZHUwgZ8wDQYJKoZIhvcNAQEBBQAD
gY0AMIGJAoGBALwCo6h4T44m+7ve+BrnEqflqBISFaZTXyJTjIVQ39ZWhE0B3Laf
bbZYju0imlQLG+MEVAtNDdiYICcBcKsapr2dxOi31Nv0moCkOj7iQueMVU4E1Tgh
YIR2I8hqixFCQIP/CMtSDail/POzFzzdVxI1pv2wRc5cL6zNwV25gbn3AgMBAAGj
ggFlMIIBYTAdBgNVHQ4EFgQUVdfBM8b6k/gnPcsgS/VajliXfXQwgcEGA1UdIwSB
uTCBtoAUVdfBM8b6k/gnPcsgS/VajliXfXShgZqkgZcwgZQxCzAJBgNVBAYTAlVT
MQswCQYDVQQIEwJXQTEhMB8GA1UEChMYVW5pdmVyc2l0eSBvZiBXYXNoaW5ndG9u
MRQwEgYDVQQLEwtVVyBTZXJ2aWNlczEXMBUGA1UEAxMOVVcgU2VydmljZXMgQ0Ex
JjAkBgkqhkiG9w0BCQEWF2hlbHBAY2FjLndhc2hpbmd0b24uZWR1ggEAMAwGA1Ud
EwQFMAMBAf8wKwYDVR0RBCQwIoYgaHR0cDovL2NlcnRzLmNhYy53YXNoaW5ndG9u
LmVkdS8wQQYDVR0fBDowODA2oDSgMoYwaHR0cDovL2NlcnRzLmNhYy53YXNoaW5n
dG9uLmVkdS9VV1NlcnZpY2VzQ0EuY3JsMA0GCSqGSIb3DQEBBAUAA4GBAIn0PNmI
JjT9bM5d++BtQ5UpccUBI9XVh1sCX/NdxPDZ0pPCw7HOOwILumpulT9hGZm9Rd+W
4GnNDAMV40wes8REptvOZObBBrjaaphDe1D/MwnrQythmoNKc33bFg9RotHrIfT4
```

```
EskaIXSx0PywbyfIR1wWxMpr8gbCjAEUHNF/
-----END CERTIFICATE-----
```

into the file and save it.

Preparing for Tomcat Configuration

The Web Server HOWTO [4] describes the configuration of the tomcat web server for secure https connections, using the certificates obtained via this document. If your `cert` directory is already located in the ~root directory on the machine hosting the web server, you are ready to proceed to the Web Server HOWTO.

If you have been working on another machine, you should prepare a tarball of the `cert` directory and transfer it to the web server host.

```
su
cd
tar cvzf cert.tgz cert
chmod 640 cert.tgz
```

To prepare for installation in tomcat, transfer the `cert.tgz` file to the ~root directory of the web server host (currently 192.168.2.2, behind the firewall).

Notes

1. https://www.washington.edu/computing/ca/index.html
2. https://certs.cac.washington.edu/
3. https://certs.cac.washington.edu/
4. ../web-server/web-server.html