Client Public/Private Key Instructions

Table of Contents

Note: In addition to the instructions below, you will need to complete the Device Certificate Request Form (DCRF) located here to include the device type, device name and file format.

Background Information

Public Key Infrastructure (PKI) protects the confidentiality of the ISO's communication with its customers.
A digital certificate is an electronic document that allows trusted digital partners to use PKI. ISO customer certificates chain to the CAISO issuing authority. Validation of the chain can assert trust between communicating parties.

A certificate is a signed public key. "Signed" refers to digital signature from the issuing authority. "Public" key infers to the existence of a private key known only to the creator of the CSR. The RIG owner creates the private key and CSR; it is not good security practice to have someone else create the private key and CSR. The RIG owner sends only the CSR to the ISO (never shares the private key).

The CSR contains the public key, mathematically linked with the private key in a way that is computationally infeasible to decode - that is, to deduce one from the other - assuming use of cryptographic algorithms expected to remain strongly resistant to attack for the duration of certificate's validity. The public key can safely be shared. The CSR contains the public key and can also be shared. The CSR is signed with the private key. The certificate authority that receives the CSR can verify the signature on the CSR by using the public key.

Generally speaking, the CSR should be created on the device, such as a RIG, with which the ISO system will securely connect.

Keys and CSRs

Depending on the type of server that you are running, you may have the ability to generate a set of server keys and certificate requests from a specific application running on your system. For example, if the system is running a Sun, IIS or Apache web server, the ability to generate the required keys and certificates are built into the product.

In other cases, your current system configuration might *not* be capable of generating keys and CSRs. If so, you may need to install and use an application that can do so. One commonly-used application is OpenSSL, which can be found on www.openssl.org. Later in this document, we detail using different tools—including OpenSSL—to create keys and CSRs.

Key Generation Requirements:

• You must use RSA keys with a key length must be a **minimum of 2048 bits.**
• Store the private key securely, i.e. using AES-256 encryption.

Certificate Signing Request (CSR) Requirements:

• The common name (CN) should reflect the server's DNS host name**.** In the case of a RIG, the common name of the RIG is provided in the RIG database documentation. Please contact EDAS@caiso.com if you need your common name provided to you.

• The CSR must be generated according to the Public Key Cryptography Standard #10 (PKCS #10).

Tools for Generating Keys and CSRs

OpenSSL

First we can create the private key. It is possible to create private key and CSR in one step, but for clarity we break out the steps:

The following command generates a 2048-bit RSA private key encrypted with the AES256 algorithm seeded by a passphrase and saved the key as keynamehere.key.

```
openssl genrsa -aes256 -out keynamehere.key 2048
```

Entering the command outputs the following, including a prompt to enter a pass phrase for your private key. You'll need that for a later step:

```
Loading 'screen' into random state - done
Generating RSA private key, 2048 bit long modulus
...........+++
...+++
unable to write 'random state'
e is 65537 (0x10001)
Enter pass phrase for keynamehere.key:
Verifying – Enter pass phrase for keynamehere.key:
```

After you create a pass phrase for your private key, you will need to enter the following command to generate a CSR from that key:

```
openssl req -new -key keynamehere.key -sha256 -out
mycsr.csr
```

You are then prompted for the pass phrase you just created for your private key in the previous step:

```
Enter pass phrase for keynamehere.key:
```

Assuming you enter in the correct password, you are then presented with the following prompt:

```
You are about to be asked to enter information that will be
incorporated into your certificate request.
What you are about to enter is what is called a
Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
```

After this prompt, you are asked to enter one-by-one various piece of information for the CSR file (country, state, city, organization, Common Name, and email address). Below, you can see examples of data you can enter:

```
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:CA
Locality Name (eg, city) []:Folsom
Organization Name (eg, company) [Internet Widgits Pty
Ltd]:California ISO
Organizational Unit Name (eg, section) []:IT
Common Name (e.g. server FQDN or YOUR name) []:RIG
Email Address []:RIG@example.com
```

Note: This creates a CSR with a DN consisting of CN, OU, O, L, ST, and C. If you want something else it will require changes to the openssl config file.

After you enter the above information, you are prompted to enter the following 'extra' attributes to be sent with your certificate request:

```
A challenge password []:
An optional company name []:
```

You have now created your CSR file.

Submit the CSR with the DCRF spreadsheet as described above to the CAISO Service Desk (ServiceDesk@caiso.com). The CAISO CA will create and sign the certificate. Before you can use the certificate, you will need to import the CAISO root and issuing CA certificates into your *keystore*. You may download this certificate chain from the CAISO Application Access page, or by following instructions in the **Receipt & Installation** section below.

Java Keytool

A system that has Java installed might have a utility called Keytool for creating and administering a *keystore*. A *keystore* is a repository of keys. In the case of using Keytool to create a CSR, we must first create a new *keystore* for that purpose alone.

The following command creates a new Java *keystore* called *myDeviceKeyStore*. It specifies RSA for the key algorithm and a keysize of 2048. The 'dname' parameter specifies the value of the key's Common Name (CN); this value will be included with the CSR and will become the certificate's CN also. You will be prompted to provide a *keystore* passphrase that you will need every time you open the *keystore* or use it to create a CSR.

```
Keytool -keystore myDeviceKeyStore -genkey -alias
myDeviceKey -keyalg RSA -keysize 2048 -dname
"cn=mydevice.example.com"
```

Now that we have created the Java *keystore*, we can use it to create a CSR. We specify that the CSR should be signed using the SHA256 algorithm with RSA.

```
Keytool -keystore myDeviceKeyStore -certreq -alias
myDeviceKey -file myDevice.csr -sigalg SHA256WithRSA
```

Submit the CSR with the DCRF spreadsheet as described above to the CAISO Service Desk (ServiceDesk@caiso.com). The CAISO CA will create and sign the certificate. Before you can use the certificate, you will need to import the CAISO root and issuing CA certificates into your *keystore*. You may download this certificate chain from the CAISO Application Access page, or by following instructions in the **Receipt & Installation** section below.

The following examples show how to use the Keytool utility to import DER-encoded certificates into your *keystore*. In this example we use *caiso_root.cer* and *caiso_issuing.cer* as the names of the certificates. The CAISO Application Access page contains multiple formats of root and issuing certificates for different purposes. There are also different certificate chains for production and test environments.

This command imports the CAISO root certificate into the *keystore*:

```
Keytool -keystore myDeviceKeyStore -import -trustcacerts -
alias CAISO_ROOT_CA -file caiso_root.cer
```

This command imports the CAISO issuing certificate into the *keystore*:

```
Keytool -keystore myDeviceKeyStore -import -trustcacerts -
alias CAISO_ISSUING_CA -file caiso_issuing.cer
```

Now, after you have received your device certificate, you can import it into your *keystore* that contains its signing authority chain; this example assumes that the device certificate you have received is called myDeviceCertificate.cer.

```
Keytool -keystore myDeviceKeyStore -import -alias
myDeviceKey -file my myDeviceCertificate.cer
```

Now you may use the Keytool utility to list all of the certificates in your *keystore*:

```
Keytool -v -list -keystore myDeviceKeyStore
```

Submittal

Once your CSR is generated, you will need to submit it along with the completed Device Certificate Request Form (DCRF) to CAISO for signing. Please forward the request to the CAISO Service Desk at ServiceDesk@caiso.com. The standard SLA associated with certificates is **ten business days** from the time the request is approved and a task is assigned to CAISO's Information Security Department to issue the certificate. Please contact the CAISO Service Desk at (888) 889-0450 if you do not receive your certificate within this timeframe.

Receipt & Installation

After receiving your signed certificate (which is essentially your certified public key), you will need to install CAISO's trusted root certificates if you have not already.

Follow these instructions to ensure you have the latest CAISO trusted root certificates:

1.) Click here to download our *CAISO Issuing Root Certificate*.
- Click Open > Install Certificate > Next
- Select the "Place all certificates in the following store" radio button
- Click Browse
- Select the "Trusted Root Certification Authorities" folder
- Click OK > Next > Finish > OK
- Click OK to close the Certificate window

2.) Click here to download our *CAISO Intermediate Issuing Certificate*.
- Click Open > Install Certificate > Next
- Verify the option "Automatically select..." radio button is selected
- Click Next>Finish>OK
- Click OK to close the Certificate window

If you are incapable of using our SHA-2 root chain for the purpose of a RIG or DPG, please contact our RIG Engineering Team at EDAS@caiso.com.