Creating RSA Keys using OpenSSL

Scott Brady
05 August 2020 · OpenSSL

Creating a private key for token signing doesn't need to be a mystery. Recently, I wrote about using OpenSSL to create keys suitable for Elliptical Curve Cryptography (ECC), and in this article, I am going to show you how to do the same for RSA private and public keys, suitable for signature generation with RSASSA-PKCS1-v1_5 and RSASSA-PSS.

tl;dr - OpenSSL RSA Cheat Sheet

```
# generate a private key with the correct length
openssl genrsa -out private-key.pem 2048

# generate corresponding public key
openssl rsa -in private-key.pem -pubout -out public-key.pem

# optional: create a self-signed certificate
openssl req -new -x509 -key private-key.pem -out cert.pem -days 360

# optional: convert pem to pfx
cat private-key.pem cert.pem > cert-with-private-key
openssl pkcs12 -export -inkey private-key.pem -in cert-with-private-key -out
```

Generating an RSA Private Key Using OpenSSL

You can generate an RSA private key using the following command:

```
openssl genrsa -out private-key.pem 2048
```

In this example, I have used a key length of 2048 bits. This is the minimum key length defined in the JOSE specs and gives you 112-bit security. This also uses an exponent of 65537, which you've likely seen serialized as "AQAB".

This gives you a PEM file containing your RSA private key, which should look something like the following:

```
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEApUy18wOzetMBY+Jw7lbXzbfTSRQbWyvIN7YbvLjfJZTF129L
DuWbRSyNd3+bNCqrOmmYFMAuKFbeGyN+fklV1MpdRaB5Ykp8r+P+ZuC4JyWhRv+J
hxT8uV0WTnKIrsg8TZHq3CvHlEE6qJYrNOA6HdrmJ01kHUo2c0KYzkNCqjuaoww8
24dRwpgqjtWkVYMF7BpVi5uLGkm+K2QO6yJhp0ZtbJxI+fOX+0g+PNC18sCzWgof
UTAjF7OGYuFps9GhdXuOE37yUJhirdEhgxGK+DSfd9Wltvq5UDvoAYWxoZTb9zAq
oubKNqOWAV1EGPzy2iJUHeEmJGH8kE3i8lBInwIDAQABAoIBAFASIqj/B/fdMnUy
AUZSpuKtwxlJNh8pzCjAfBsk6mMH9/XtoUwsCNSvSi+yjnnsmVkIQXT7yuAbhCdd
QC7oUz1qcVgC7gmgz1lcdaVcAZhk8AS2T+YxUmJwJxgE/xS7RgrFPiE8y8aS+lkj
tPY+D6jamlY7dN2DT3Dxt5dimW5f/aRguqBcM/Eg+8K0HK6h43A6usL/J9WJwXQv
```

Creating an RSA Public Key from a Private Key Using OpenSSL

Now that you have your private key, you can use it to generate another PEM file, containing only your public key.

```
openssl rsa -in private-key.pem -pubout -out public-key.pem
```

This should give you another PEM file, containing the public key:

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEApUy18wOzetMBY+Jw7lbX
zbfTSRQbWyvIN7YbvLjfJZTF129LDuWbRSyNd3+bNCqrOmmYFMAuKFbeGyN+fklV
1MpdRaB5Ykp8r+P+ZuC4JyWhRv+JhxT8uV0WTnKIrsg8TZHq3CvHlEE6qJYrNOA6
HdrmJ01kHUo2c0KYzkNCqjuaoww824dRwpgqjtWkVYMF7BpVi5uLGkm+K2QO6yJh
p0ZtbJxI+fOX+0g+PNC18sCzWgofUTAjF7OGYuFps9GhdXuOE37yUJhirdEhgxGK
+DSfd9Wltvq5UDvoAYWxoZTb9zAqoubKNqOWAV1EGPzy2iJUHeEmJGH8kE3i8lBI
nwIDAQAB
-----END PUBLIC KEY-----
```

Creating an RSA Self-Signed Certificate Using OpenSSL

Now that you have a private key, you can use it to generate a self-signed certificate. This is not required, but it allows you to use the key for server/client authentication, or gain X509 specific functionality in technologies such as JWT and SAML.

```
openssl req -new -x509 -key private-key.pem -out cert.pem -days 360
```

This will again generate yet another PEM file, this time containing the certificate created by your private key:

```
lNv3MCqi5so2o5YBXUQY/PLaIlQd4SYkYfyQTeLyUEifAgMBAAGjUzBRMB0GA1Ud
DgQWBBT/T4XbpUQiaDCdoXIiSOm7fscYXjAfBgNVHSMEGDAWgBT/T4XbpUQiaDCd
oXIiSOm7fscYXjAPBgNVHRMBAf8EBTADAQH/MA0GCSqGSIb3DQEBCwUAA4IBAQCO
viX8cTrj8pKGtFojDPAo31F+/5WgOAhTLmHIOyOrGYEz5uf2BKI0pNFxX7MbBJJD
s6hqpf0Y1RVyhW6RPZGg3H2jnTrnTJBaXuf+mYWnekobMmalCSjXG6a64nsC8lc/
4jfNtr7cejMuYx918DFRs4Uhlm7v3LSn2dqonv5xN633ou6fDyH6MA/G6D1TO4LO
OK1QXd0ljSldwLPfLia3X10DgddyqL4kb2Xcy4i9Nmve//72WMzvsgymdXJLTGOa
dM/rUby2k+XzeKjeU0mIuBpv1aRPkKq22TsScdMF4YE3cR/3JaIkSKvOV8fY6m/F
giUzuDlljDZxARq+GLKh
-----END CERTIFICATE-----
```

You could leave things there, but often, when working on Windows, you will need to create a PFX file that contains both the certificate and the private key for you to export and use.

You can do this by first concatenating your private key and certificate into a single file:

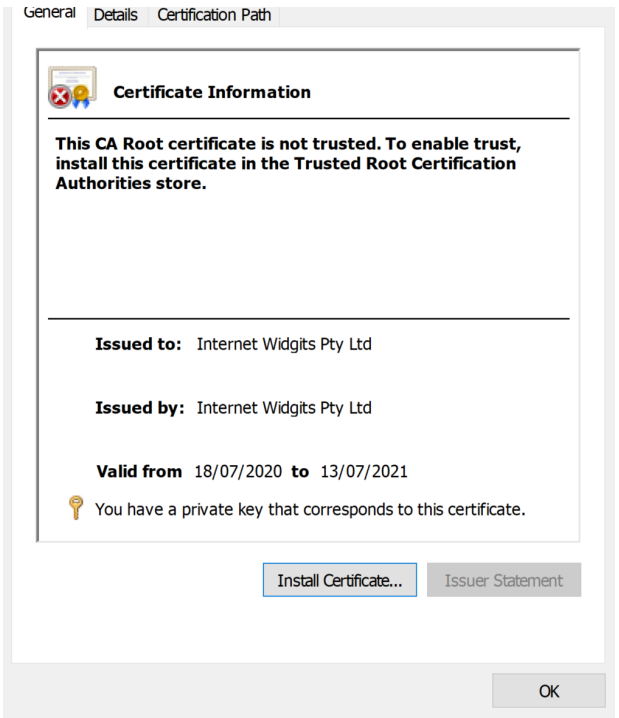
```
cat private-key.pem cert.pem > cert-with-private-key
```

And then using OpenSSL to create a PFX file:

```
openssl pkcs12 -export -inkey private-key.pem -in cert-with-private-key -out
```

OpenSSL will ask you to create a password for the PFX file. Feel free to leave this blank.

This should leave you with a certificate that Windows can both install and export the RSA private key from.



Scott Brady

I'm the Identity & Access Control Lead at Rock Solid Knowledge; a software developer focusing on authentication, FIDO2, OAuth, and OpenID Connect.

About Upcoming Talks Pluralsight Courses Consultancy

2 Comments ScottBrady91 Disqus' Privacy Policy 1 Login

Recommend Tweet Share Sort by Newest

Join the discussion…

LOG IN WITH OR SIGN UP WITH DISQUS ?

Name

Desmond Koh • 5 months ago • edited

According to OpenSSL docs (https://wiki.openssl.org/in..., `genrsa` has been superceded by `genpkey`. Any diiference?

• Reply • Share ›

Scott Brady Mod Desmond Koh • 4 months ago

It seems to be a case of genpkey being a more generic version that handles more than just RSA. It also outputs the key using the PKCS#8 format, rather than the PKCS#1 format: https://unix.stackexchange..... I'll take a look into it and update this article if necessary.

• Reply • Share ›

Subscribe Add Disqus to your siteAdd DisqusAdd Do Not Sell My Data

© 2015 - 2021 Scott Brady | Privacy & Licensing