How to Load RSA Private Key From File

Asked 14 years ago Modified 4 years, 1 month ago Viewed 125k times

64

I am working on a test harness for a SAML 1.1 Assertion Consumer Service. The test must generate a signed SAMLResponse and submit it to the ACS encoded in Base64. The ACS must be able to verify the signed message using the X509 public cert.

I am able to build the SAMLResponse, adding the necessary assertions, etc. But when I try to sign the object I am running into problems. Here is a snippet of my current code:

```
String certPath = "mycert.pem";
File pubCertFile = new File(certPath);
BufferedInputStream bis = null;
try {
    bis = new BufferedInputStream(new FileInputStream(pubCertFile));
} catch(FileNotFoundException e) {
    throw new Exception("Could not locate certfile at '" + certPath + "'", e);
}
CertificateFactory certFact = null;
Certificate cert = null;
try {
    certFact = CertificateFactory.getInstance("X.509");
    cert = certFact.generateCertificate(bis);
} catch(CertificateException e) {
    throw new Exception("Could not instantiate cert", e);
}
bis.close();
ArrayList<Certificate> certs = new ArrayList<Certificate>();
certs.add(cert);

String keyPath = "mykey.pem";
File privKeyFile = new File(keyPath);
try {
    bis = new BufferedInputStream(new FileInputStream(privKeyFile));
} catch(FileNotFoundException e) {
    throw new Exception("Could not locate keyfile at '" + keyPath + "'", e);
}
byte[] privKeyBytes = new byte[(int)privKeyFile.length()];
bis.read(privKeyBytes);
bis.close();
KeyFactory keyFactory = KeyFactory.getInstance("RSA");
KeySpec ks = new PKCS8EncodedKeySpec(privKeyBytes);
RSAPrivateKey privKey = (RSAPrivateKey) keyFactory.generatePrivate(ks);

samlResponse.sign(Signature.getInstance("SHA1withRSA").toString(), privKey,
certs);
```

The error occurs on the second-to-last line. I see the following in the console:

```
java.security.spec.InvalidKeySpecException: java.security.InvalidKeyException:
invalid key format
```

Though not customary or secure, but for sake of this thread, I am providing the public cert and private key that I am using. I of course will re-create new ones once the problem is solved. :)

```
aj@mmdev0:~/$ cat mykey.pem
-----BEGIN RSA PRIVATE KEY-----
MIICXgIBAAKBgQDnbcLSlDFaDMhalcmQgclTFobpkHQHJtxMVGRlbv7zknttAVbY
1jzGjJ6HVupndzDxA9tbiMjQujmGlS/8g5IEbVsR9o6dmcmbvujtEZ2rHZ82tMYP
VAt2IoS/W/q2Rr1cAZ/zTKEmh0ZZjzCZFueLfrYPm3am5JLcXgVtbKwybQIDAQAB
AoGBAJ441oettYgBUUFNQv8/HGtn7Vjl38277cVptTH8DuZr8WJ3Fe8tmWONZBzX
eW6/eIBuyJvuCo1ZpFa0zJfxQ/Ph6QlQwdN50GNfh9RzSS6lDdfy8BRhc27sypXS
L6c5ljB6ql+pp3DdxFhJMOs3ZmBJdeyWe7uFrkngtnM1nxZBAkEA+1hbV1Q305wa
u8YMF1SlNIAfgLJ7buD43SEXle0egz405PFG8f8yDmvROwDiRceILGVrRbInd7Cb
dvJKr34WOQJBAOu2+reG44rNuiXeGX1MYg6TlWYyABm7PrTrhPZkedodOQB8p7zD
AqtDSK7RnDCoThndPW6kdNAeB+kG4ug5XdUCQHRDU8UajNRSkj8nhjJIkj6twWS7
qsMIR7Wp+An+7C1TWg5I2UNZg2MOVnNPnlseyAuZQjy0AvOnetJTk16IGWkCQQCL
FUbOr8rnhgiGe4yywDVDwJVw3aPtiuyvOCEWeabkqkWOIf+fg7m5cFQcwxXUKBsd
a8vp0yQSAQZN24Bb4i2ZAkEA8xGJFlFDY9HREWZnDey5STgbUeT1wYkyKcDsUrp1
kR/3BliGqSIfje+mSKDIZqaP+gai/8bIABYAsDP/t6+cuA==
-----END RSA PRIVATE KEY-----

aj@mmdev0:~/$ cat mycert.pem
-----BEGIN CERTIFICATE-----
MIID7zCCA1igAwIBAgIJAKrURaAaD6ulMA0GCSqGSIb3DQEBBQUAMIGsMQswCQYD
VQQGEwJVUzERMA8GA1UECBMISWxsaW5vaXMxEDAOBgNVBAcTB0NoaWNhZ28xHDAa
BgNVBAoTE0hvc3R3YXkgQ29ycG9yYXRpb24xITAfBgNVBAsTGFJlc2VhcmNoIGFu
ZCBEZXZlbG9wbWVudDEYMBYGA1UEAxMPd3d3Lmhvc3R3YXkuY29tMR0wGwYJKoZI
hvcNAQkBFg5hakBob3N0d2F5LmNvbTAeFw0xMDA3MTQwMjMyMDhaFw0xMTA3MTQw
MjMyMDhaMIGsMQswCQYDVQQGEwJVUzERMA8GA1UECBMISWxsaW5vaXMxEDAOBgNV
BAcTB0NoaWNhZ28xHDAaBgNVBAoTE0hvc3R3YXkgQ29ycG9yYXRpb24xITAfBgNV
BAsTGFJlc2VhcmNoIGFuZCBEZXZlbG9wbWVudDEYMBYGA1UEAxMPd3d3Lmhvc3R3
YXkuY29tMR0wGwYJKoZIhvcNAQkBFg5hakBob3N0d2F5LmNvbTCBnzANBgkqhkiG
9w0BAQEFAAOBjQAwgYkCgYEA523C0pQxWgzIWpXJkIHJUxaG6ZB0BybcTFRkZW7+
85J7bQFW2NY8xoyeh1bqZ3cw8QPbW4jI0Lo5hpUv/IOSBG1bEfaOnZnJm77o7RGd
qx2fNrTGD1QLdiKEv1v6tka9XAGf80yhJodGWY8wmRbni362D5t2puSS3F4FbWys
Mm0CAwEAAaOCARUwggERMB0GA1UdDgQWBBQI/4Inzs6OH5IquItuKhIrhPb24zCB
4QYDVR0jBIHZMIHWgBQI/4Inzs6OH5IquItuKhIrhPb246GBsqSBrzCBrDELMAkG
A1UEBhMCVVMxETAPBgNVBAgTCElsbGlub2lzMRAwDgYDVQQHEwdDaGljYWdvMRww
GgYDVQQKExNIb3N0d2F5IENvcnBvcmF0aW9uMSEwHwYDVQQLExhSZXNlYXJjaCBh
bmQgRGV2ZWxvcG1lbnQxGDAWBgNVBAMTD3d3dy5ob3N0d2F5LmNvbTEdMBsGCSqG
SIb3DQEJARYOYWpAaG9zdHdheS5jb22CCQCq1EWgGg+rpTAMBgNVHRMEBTADAQH/
MA0GCSqGSIb3DQEBBQUAA4GBAA388zZp6UNryC/6o44hj7wTBQdzFFM5cs3B668A
ylAnnal+J8RMIeCHoMF4S7yFQtYdOiWeScgw3c7KXrhJK1X7fU3I+eb1t3Yp1cTI
htyzw14AoiICFalmlVgTCsn3+uh6AXP02PTkR8osdEpUOlWap4uzSKYNKc7tLOFd
4CkM
-----END CERTIFICATE-----
```

Thanks!

java rsa saml

Share Edit Follow

edited Jul 14, 2010 at 15:25

asked Jul 14, 2010 at 2:34

AJ.

28k ● 19 ● 85 ● 95

See my post in <stackoverflow.com/questions/51706391/...> – Nicola De Nisco Jan 27, 2021 at 12:35

2 Answers

Sorted by: Highest score (default)

78

You need to convert your private key to PKCS8 format using following command:

```
openssl pkcs8 -topk8 -inform PEM -outform DER -in private_key_file  -nocrypt >
pkcs8_key
```

After this your java program can read it.

Share Edit Follow

answered Sep 19, 2011 at 16:02

Alexander Kuznetsov

3,092 ● 2 ● 26 ● 30

1 the reason is that the PKCS8 encoding adds the algorithm identifier. The earlier value had "RSA key" in the informal header – user1778602 Mar 26, 2018 at 21:38

Is this meant to output a file that I can't read in a text editor anymore? – minseong Mar 9, 2021 at 0:03

21

Two things. First, you must base64 decode the `mykey.pem` file yourself. Second, the openssl private key format is specified in [PKCS#1](https://) as the `RSAPrivateKey` ASN.1 structure. It is not compatible with java's `PKCS8EncodedKeySpec`, which is based on the `SubjectPublicKeyInfo` ASN.1 structure. If you are willing to use the bouncycastle library you can use a few classes in the bouncycastle provider and bouncycastle PKIX libraries to make quick work of this.

```java
import java.io.BufferedReader;
import java.io.FileReader;
import java.security.KeyPair;
import java.security.Security;

import org.bouncycastle.jce.provider.BouncyCastleProvider;
import org.bouncycastle.openssl.PEMKeyPair;
import org.bouncycastle.openssl.PEMParser;
import org.bouncycastle.openssl.jcajce.JcaPEMKeyConverter;

// ...
```

```
String keyPath = "mykey.pem";
BufferedReader br = new BufferedReader(new FileReader(keyPath));
Security.addProvider(new BouncyCastleProvider());
PEMParser pp = new PEMParser(br);
PEMKeyPair pemKeyPair = (PEMKeyPair) pp.readObject();
KeyPair kp = new JcaPEMKeyConverter().getKeyPair(pemKeyPair);
pp.close();
samlResponse.sign(Signature.getInstance("SHA1withRSA").toString(),
kp.getPrivate(), certs);
```

Share Edit Follow

edited Oct 7, 2021 at 7:59
Community Bot
1 ● 1

answered Jul 15, 2010 at 0:36
President James K. Polk
41.4k ● 24 ● 98 ● 129

5 Thanks for the tip Greg. I'm ok with using BC (I am already using it to sign keys), but I would like to know how to decode a private key from DER using the default provider. It's easy enough to encode, decoding doesn't appear to be so straightforward... I encoded the key by calling key.getEncoded(). Easy as falling off a log. BTW, the key was created using keytool. I was thinking I could decode using an "RSA" key factory. There is a generatePrivate() method that returns the private key I want, but that method requires a KeySpec. The RSAPrivateKeySpec takes a modulus and private exponent. Is th – user619890 Feb 16, 2011 at 15:12

Still no solution? I have the same problem. (this should not have been flagged as the correct solution if it is not the solution, by the way). – Vincent Cantin Aug 12, 2011 at 12:56

1 @Vincent: I believe it was the solution to the original question. It may not have answered the additional question in the comment, but that comment was garbled at the end. Also, the comment was made 7 months after my answer!. You say you have the same problem. Can you be more specific? – President James K. Polk Aug 13, 2011 at 15:44

I found the solution myself already, the bounty is kind of obsolete now. – Vincent Cantin Aug 18, 2011 at 15:36

2 @Vincent: You never said what your problem was anyway. – President James K. Polk Aug 18, 2011 at 20:26