

Schlüsselpaar generieren

Inhaltsverzeichnis

- [1 Schlüsselgenerierung](#)
 - [1.1 Online Service](#)
 - [1.2 Lokale Generierung](#)
- [2 Umgang mit Schlüsseln](#)

Der sichere Zugangsdatenspeicher benötigt ein individuelles Schlüsselpaar, welches zur Ver- und Entschlüsselung benötigt wird.



Photo by Uwe Baumann on pixabay

Alle sensiblen Daten und Kennwörter werden mit einem symmetrischen Schlüsselpaar verschlüsselt. Die Sicherheit des Zugangsdatenspeicher besteht darin, dass der private Schlüssel nicht am Server gespeichert wird. Dieser muss vor der Entschlüsselung hochgeladen werden, um die Daten zu entschlüsseln.

1 Schlüsselgenerierung

Dieses Schlüsselpaar muss nach der Installation erstellt werden. Es gibt dafür mehrere Möglichkeiten. Es wird ein RSA Schlüssel im PEM-Format (base64-kodiert) mit einer empfohlenen Schlüssellänge von 4096 bit benötigt.

1.1 Online Service

Der einfachste (wenn auch nicht sicherste Weg) ist die Generierung über einen kostenlosen Online-Service wie [JSCrypt](#). Wählen Sie hier eine Key Size von 4096 bit aus und klicken anschließend auf Generate New Keys.

Online RSA Key Generator

Key Size

4096 bit

Generate New Keys

Generated in 15354 ms

☐ Async

Private Key

-----BEGIN RSA PRIVATE KEY-----
MIIJwIBAAKCAgBtIFLO0IPZPtnCCS4yYcHZXaQvX3597I0vNpmo0/6fmMvGgMO
StdIgoIhuDovLMifMpC8dm25ICJEprCacUK0GCC5SofyvWkMhINPZ5nWALUw8I
dJM
xYNa+AOOvJMI8WmDmWwIBskmNPlx2K0a3VPHF3ZaA3OT02L6oOgYPnEFkf
B8Vtkg
I6OPCNx/8jCezntElngGoktH/BjGBt7m6TQzkVKgvUT3hvcGfktJ0+VJjeA67kRm
66/pBQX/b5ERh8r4ITxt1KfPI0mRJtob9RjEJmWtMEauV0MdQIDe5BXsVKEHJ
Vn
t1AZkutorJxRrPEWmOyyLO31lgABt6V7uFJf5d9HleMC6iYJ80gQcJQcJPurJi
68LRicaJAq/Bao2yEFrXdFoLzJtBcj92W16Y8gVXi9/N1RP3AddxGLBseNp2YD
Zp
xde/ayNN7yw6aJxXcuP/M53D0cT7O0oFSsUDMS2oJSFzL96xi+ansH9Z53Tm
nEVa

Public Key

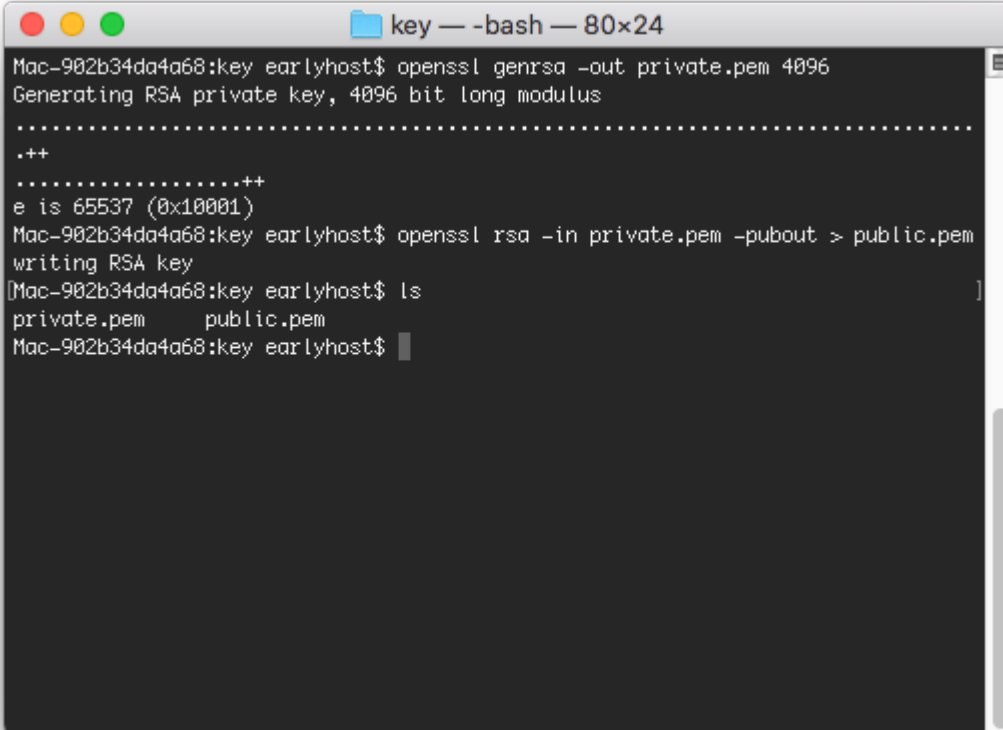
-----BEGIN PUBLIC KEY-----
MIICITANBgkqhkiG9w0BAQEFAAOCAg4AMIICQKCAgBtIFLO0IPZPtnCCS4
yYcH
ZXaQvX3597I0vNpmo0/6fmMvGgMOSfdIgoIhuDovLMifMpC8dm25ICJEprCac
UK0
GCC5SofyvWkMhINPZ5nWALUw8IdjMxYNa+AOOvJMI8WmDmWwIBskmNPlx2
K0a3VPH
F3ZaA3OT02L6oOgYPnEFkfB8VtkgI6OPCNx/8jCezntElngGoktH/BjGBt7m6TQ
z
kVKgvUT3hvcGfktJ0+VJjeA67kRm66/pBQX/b5ERh8r4ITxt1KfPI0mRJtob9Rj
EJmWtMEauV0MdQIDe5BXsVKEHJVnt1AZkutorJxRrPEWmOyyLO31lgABt6V7
uFJ
f5d9HleMC6iYJ80gQcJQcJPurJi68LRicaJAq/Bao2yEFrXdFoLzJtBcj92W16Y
8gVXi9/N1RP3AddxGLBseNp2YDZpxde/ayNN7yw6aJxXcuP/M53D0cT7O0oF
SsUD

1.2 Lokale Generierung

Der sicherste Weg ist das Schlüsselpaar lokal zu generieren, da der private Schlüssel potentiell mit keiner fremden Webseite geteilt wird. Es gibt dafür unterschiedliche Tools wie `openssl`, welches auf den meisten Linux und MacOS Systemen bereits vorinstalliert ist. Führen Sie folgende Befehle aus.

Code

```
openssl genrsa -out private.pem 4096
openssl rsa -in private.pem -pubout > public.pem
```



```
key — -bash — 80x24
Mac-902b34da4a68:key earlyhost$ openssl genrsa -out private.pem 4096
Generating RSA private key, 4096 bit long modulus
.....
.++
.....++
e is 65537 (0x10001)
Mac-902b34da4a68:key earlyhost$ openssl rsa -in private.pem -pubout > public.pem
writing RSA key
Mac-902b34da4a68:key earlyhost$ ls
private.pem  public.pem
Mac-902b34da4a68:key earlyhost$
```

Die Datei `private.pem` beinhaltet Ihren privaten Schlüssen. Die Datei `public.pem` den dazugehörigen öffentlichen Schlüssel.

2 Umgang mit Schlüsseln

Der öffentliche Schlüssel muss im ACP unter `Inhalte > Zugangsdaten-Speicher > Öffentliche Schlüssel` hinterlegt werden. Die Zugangsdaten werden ab sofort mit dem öffentlichen Schlüssel verschlüsselt und können nur mit dem privaten Schlüssel entschlüsselt werden. Speichern Sie diesen Schlüssel sicher ab. Wenn Sie ihn verlieren können Sie nicht auf die verschlüsselten Zugangsdaten zugreifen. Zugangsdaten, welche vor dem Hinzufügen des öffentlichen Schlüssels erstellt wurden, können nicht entschlüsselt werden.

