RSA keys

```
openssl genrsa -des3 -out private.pem 2048
```

```
openssl rsa -in private.pem -outform PEM -pubout -out public.pem
```

Export private key (unencrypted!)

```
openssl rsa -in private.pem -out private_unencrypted.pem -outform PEM
```

Export pub key in OpenSSH format:

```
ssh-keygen -y -f private.pem
```

CA Bundle

Extract CAs form Mozilla

Direct download link cacert.pem
Page https://curl.haxx.se/docs/caextract.html

Add own CA cert

```
sudo cp foo.crt /usr/local/share/ca-certificates/foo.crt
sudo update-ca-certificates
```

Info

Nice site verification tool: SSL Labs

```
openssl s_client -showcerts -connect smtp.gmail.com:587 -starttls smtp
openssl s_client -connect host.host:9999

# With HTTP server name:
openssl s_client -connect host.host:9999 -servername myhostname.domain.com

openssl x509 -in certificate.pem -text
```

Verify crt, csr and key

```
openssl x509 -noout -modulus -in certificate.crt | openssl md5
openssl rsa -noout -modulus -in privateKey.key | openssl md5
```

```
openssl req -noout -modulus -in CSR.csr | openssl md5
```

Generate key

```
openssl dhparam -dsaparam -out dh2048.pem 2048
openssl genrsa -des3 -out domain.com.key 2048
```

Remove password from keyfile:

```
openssl rsa -in www.key -out new.key
```

Generate CSR

```
openssl req -new -key domain.com.key -out wild.domain.com.csr
```

With SHA256

```
openssl req -new -key domain.com.key -out wild.domain.com.csr -sha256
```

Server certificate chain

RFC 4346

```
  certificate_list
    This is a sequence (chain) of X.509v3 certificates.  The sender's
    certificate must come first in the list.  Each following
    certificate must directly certify the one preceding it.  Because
    certificate validation requires that root keys be distributed
    independently, the self-signed certificate that specifies the root
    certificate authority may optionally be omitted from the chain,
    under the assumption that the remote end must already possess it
    in order to validate it in any case.
```

It is required to put not only site certificate in your web server configuration, but also provide intermediate certificate chain. If your server certificate is in PEM format (text), additional certificates can be simply concatenated. All certificates should be in correct order. To verify order

```
openssl s_client -connect gmail.com:443 -servername gmail.com
```

```
Certificate chain
 0 s:/C=US/ST=California/L=Mountain View/O=Google Inc/CN=gmail.com
   i:/C=US/O=Google Inc/CN=Google Internet Authority G2
 1 s:/C=US/O=Google Inc/CN=Google Internet Authority G2
   i:/C=US/O=GeoTrust Inc./CN=GeoTrust Global CA
 2 s:/C=US/O=GeoTrust Inc./CN=GeoTrust Global CA
   i:/C=US/O=Equifax/OU=Equifax Secure Certificate Authority
```