SSL Certificates Cheat-Sheet

X.509 is an ITU standard defining the format of public key certificates. X.509 are used in TLS/SSL, which is the basis for HTTPS. An X.509 certificate binds an identity to a public key using a digital signature. A certificate contains an identity (hostname, organization, etc.) and a public key (RSA, DSA, ECDSA, ed25519, etc.), and is either signed by a Certificate Authority or is Self-Signed.

Self-Signed Certificates

Generate CA

1. Generate RSA

```
openssl genrsa -aes256 -out ca-key.pem 4096
```

2. Generate a public CA Cert

```
openssl req -new -x509 -sha256 -days 365 -key ca-key.pem -out ca.pem
```

Generate Certificate

1. Create a RSA key

```
openssl genrsa -out cert-key.pem 4096
```

2. Create a Certificate Signing Request (CSR)

```
openssl req -new -sha256 -subj "/CN=yourcn" -key cert-key.pem -out
cert.csr
```

3. Create a `extfile` with all the alternative names

```
echo "subjectAltName=DNS:your-dns.record,IP:257.10.10.1" >> extfile.cnf
```

```
# optional
echo extendedKeyUsage = serverAuth >> extfile.cnf
```

4. Create the certificate

```
openssl x509 -req -sha256 -days 365 -in cert.csr -CA ca.pem -CAkey ca-
key.pem -out cert.pem -extfile extfile.cnf -CAcreateserial
```

5. Combine both certificates into one

```
cat cert.pem > fullchain.pem
cat ca.pem >> fullchain.pem
```

Certificate Formats

X.509 Certificates exist in Base64 Formats **PEM (.pem, .crt, .ca-bundle)**, **PKCS#7 (.p7b, p7s)** and Binary Formats **DER (.der, .cer)**, **PKCS#12 (.pfx, p12)**.

Convert Certs

COMMAND	CONVERSION
`openssl x509 -outform der -in cert.pem -out cert.der`	PEM to DER
`openssl x509 -inform der -in cert.der -out cert.pem`	DER to PEM
`openssl pkcs12 -in cert.pfx -out cert.pem -nodes`	PFX to PEM

Verify Certificates

`openssl verify -CAfile ca.pem -verbose cert.pem`

Install the CA Cert as a trusted root CA

On Debian & Derivatives

- Move the CA certificate (`ca.pem`) into `/usr/local/share/ca-certificates/ca.crt`.
- Update the Cert Store with:

```
sudo update-ca-certificates
```

Refer the documentation here and here.

On Fedora

- Move the CA certificate (`ca.pem`) to `/etc/pki/ca-trust/source/anchors/ca.pem` or `/usr/share/pki/ca-trust-source/anchors/ca.pem`
- Now run (with sudo if necessary):

```
update-ca-trust
```

Refer the documentation here.

On Arch

System-wide – Arch(p11-kit)
(From arch wiki)

- Run (As root)

```
trust anchor --store myCA.crt
```

- The certificate will be written to /etc/ca-certificates/trust-source/myCA.p11-kit and the "legacy" directories automatically updated.
- If you get "no configured writable location" or a similar error, import the CA manually:
- Copy the certificate to the /etc/ca-certificates/trust-source/anchors directory.
- and then

```
update-ca-trust
```

wiki page here

On Windows

Assuming the path to your generated CA certificate as `C:\ca.pem`, run:

```
Import-Certificate -FilePath "C:\ca.pem" -CertStoreLocation
Cert:\LocalMachine\Root
```

- Set `-CertStoreLocation` to `Cert:\CurrentUser\Root` in case you want to trust certificates only for the logged in user.

OR

In Command Prompt, run:

```
certutil.exe -addstore root C:\ca.pem
```

- `certutil.exe` is a built-in tool (classic `System32` one) and adds a system-wide trust anchor.

On Android

The exact steps vary device-to-device, but here is a generalised guide:

1. Open Phone Settings
2. Locate `Encryption and Credentials` section. It is generally found under `Settings > Security > Encryption and Credentials`
3. Choose `Install a certificate`
4. Choose `CA Certificate`
5. Locate the certificate file `ca.pem` on your SD Card/Internal Storage using the file manager.
6. Select to load it.
7. Done!