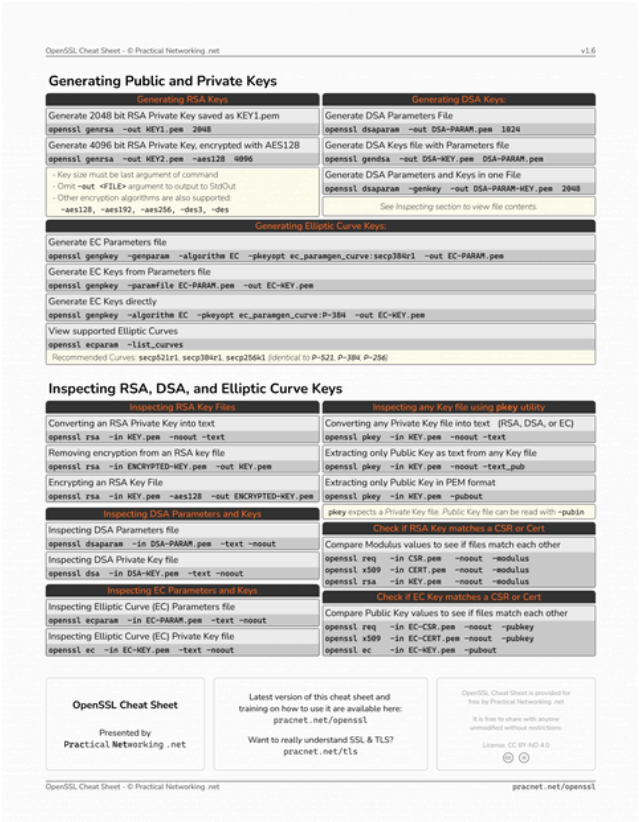
OpenSSL Training

OpenSSL Training - Based on the OpenSSL Cheat Sheet



OpenSSL Cheat Sheet - © Practical Networking .net v1.6

Generating Public and Private Keys

Generating RSA Keys	Generating DSA Keys:
Generate 2048 bit RSA Private Key saved as KEY1.pem `openssl genrsa -out KEY1.pem 2048`	Generate DSA Parameters File `openssl dsaparam -out DSA-PARAM.pem 1024`
Generate 4096 bit RSA Private Key, encrypted with AES128 `openssl genrsa -out KEY2.pem -aes128 4096`	Generate DSA Keys file with Parameters file `openssl gendsa -out DSA-KEY.pem DSA-PARAM.pem`
- Key size must be last argument of command - Omit `-out <FILE>` argument to output to StdOut - Other encryption algorithms are also supported: `-aes128, -aes192, -aes256, -des3, -des`	Generate DSA Parameters and Keys in one File `openssl dsaparam -genkey -out DSA-PARAM-KEY.pem 2048` *See Inspecting section to view file contents.*

Generating Elliptic Curve Keys:
Generate EC Parameters file `openssl genpkey -genparam -algorithm EC -pkeyopt ec_paramgen_curve:secp384r1 -out EC-PARAM.pem`
Generate EC Keys from Parameters file `openssl genpkey -paramfile EC-PARAM.pem -out EC-KEY.pem`
Generate EC Keys directly `openssl genpkey -algorithm EC -pkeyopt ec_paramgen_curve:P-384 -out EC-KEY.pem`
View supported Elliptic Curves `openssl ecparam -list_curves` Recommended Curves: `secp521r1`, `secp384r1`, `secp256k1` (identical to *P-521*, *P-384*, *P-256*)

Inspecting RSA, DSA, and Elliptic Curve Keys

Inspecting RSA Key Files	Inspecting any Key file using pkey utility
Converting an RSA Private Key into text `openssl rsa -in KEY.pem -noout -text`	Converting any Private Key file into text (RSA, DSA, or EC) `openssl pkey -in KEY.pem -noout -text`
Removing encryption from an RSA key file `openssl rsa -in ENCRYPTED-KEY.pem -out KEY.pem`	Extracting only Public Key as text from any Key file `openssl pkey -in KEY.pem -noout -text_pub`
Encrypting an RSA Key File `openssl rsa -in KEY.pem -aes128 -out ENCRYPTED-KEY.pem`	Extracting only Public Key in PEM format `openssl pkey -in KEY.pem -pubout`
Inspecting DSA Parameters and Keys	`pkey` expects a Private Key file. Public Key file can be read with `-pubin`
Inspecting DSA Parameters file `openssl dsaparam -in DSA-PARAM.pem -text -noout`	Check if RSA Key matches a CSR or Cert

Inspecting DSA Private Key file
`openssl dsa -in DSA-KEY.pem -text -noout`
	Compare Modulus values to see if files match each other
	`openssl req -in CSR.pem -noout -modulus`
	`openssl x509 -in CERT.pem -noout -modulus`
	`openssl rsa -in KEY.pem -noout -modulus`

Inspecting EC Parameters and Keys	Check if EC Key matches a CSR or Cert

Inspecting Elliptic Curve (EC) Parameters file
`openssl ecparam -in EC-PARAM.pem -text -noout`
	Compare Public Key values to see if files match each other
	`openssl req -in EC-CSR.pem -noout -pubkey`
	`openssl x509 -in EC-CERT.pem -noout -pubkey`
	`openssl ec -in EC-KEY.pem -pubout`

Inspecting Elliptic Curve (EC) Private Key file `openssl ec -in EC-KEY.pem -text -noout`	

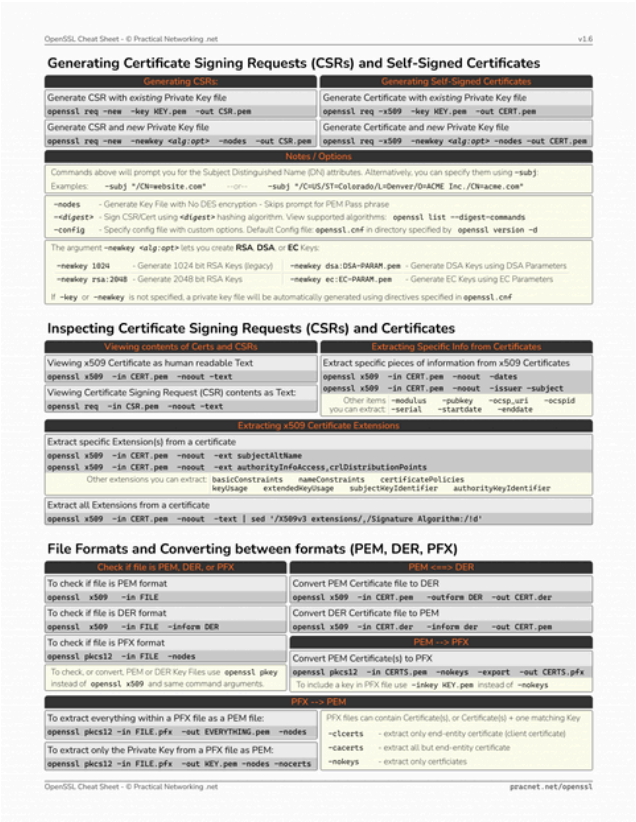
OpenSSL Cheat Sheet

Presented by Practical Networking .net

Latest version of this cheat sheet and training on how to use it are available here: pracnet.net/openssl

Want to really understand SSL & TLS? pracnet.net/tls

OpenSSL Cheat Sheet - © Practical Networking .net pracnet.net/openssl



OpenSSL Cheat Sheet - © Practical Networking .net v1.6

Generating Certificate Signing Requests (CSRs) and Self-Signed Certificates

Generating CSRs:	Generating Self-Signed Certificates
Generate CSR with existing Private Key file `openssl req -new -key KEY.pem -out CSR.pem`	Generate Certificate with existing Private Key file `openssl req -x509 -key KEY.pem -out CERT.pem`
Generate CSR and *new* Private Key file `openssl req -new -newkey <alg:opt> -nodes -out CSR.pem`	Generate Certificate and *new* Private Key file `openssl req -x509 -newkey <alg:opt> -nodes -out CERT.pem`

Notes / Options
Commands above will prompt you for the Subject Distinguished Name (DN) attributes. Alternatively, you can specify them using `-subj` Examples: `-subj "/CN=website.com"` -or- `-subj "/C=US/ST=Colorado/L=Denver/O=ACME Inc./CN=acme.com"`

`-nodes` - Generate Key File with No DES encryption - Skips prompt for PEM Pass phrase
`-<digest>` - Sign CSR/Cert using `<digest>` hashing algorithm. View supported algorithms: `openssl list --digest-commands`
`-config` - Specify config file with custom options. Default Config file: `openssl.cnf` in directory specified by `openssl version -d`

The argument `-newkey <alg:opt>` lets you create **RSA**, **DSA**, or **EC** Keys: `-newkey 1024` - Generate 1024 bit RSA Keys (legacy) `-newkey rsa:2048` - Generate 2048 bit RSA Keys `-newkey dsa:DSA-PARAM.pem` - Generate DSA Keys using DSA Parameters `-newkey ec:EC-PARAM.pem` - Generate EC Keys using EC Parameters If `-key` or `-newkey` is not specified, a private key file will be automatically generated using directives specified in `openssl.cnf`

Inspecting Certificate Signing Requests (CSRs) and Certificates

Viewing contents of Certs and CSRs	Extracting Specific Info from Certificates

Viewing x509 Certificate as human readable Text
`openssl x509 -in CERT.pem -noout -text`
	Extract specific pieces of information from x509 Certificates
	`openssl x509 -in CERT.pem -noout -dates`
	`openssl x509 -in CERT.pem -noout -issuer -subject`
	Other items you can extract: `-modulus -pubkey -ocsp_uri -ocspid -serial -startdate -enddate`

Viewing Certificate Signing Request (CSR) contents as Text: `openssl req -in CSR.pem -noout -text`	

Extracting x509 Certificate Extensions

Extract specific Extension(s) from a certificate
`openssl x509 -in CERT.pem -noout -ext subjectAltName`
`openssl x509 -in CERT.pem -noout -ext authorityInfoAccess,crlDistributionPoints`
Other extensions you can extract: `basicConstraints nameConstraints certificatePolicies keyUsage extendedKeyUsage subjectKeyIdentifier authorityKeyIdentifier`

Extract all Extensions from a certificate `openssl x509 -in CERT.pem -noout -text | sed '/X509v3 extensions:/,/Signature Algorithm:/!d'`

File Formats and Converting between formats (PEM, DER, PFX)

Check if file is PEM, DER, or PFX	PEM <==> DER
To check if file is PEM format `openssl x509 -in FILE`	Convert PEM Certificate file to DER `openssl x509 -in CERT.pem -outform DER -out CERT.der`
To check if file is DER format `openssl x509 -in FILE -inform DER`	Convert DER Certificate file to PEM `openssl x509 -in CERT.der -inform der -out CERT.pem`
To check if file is PFX format `openssl pkcs12 -in FILE -nodes`	PEM --> PFX
To check, or convert, PEM or DER Key Files use `openssl pkey` instead of `openssl x509` and same command arguments.	Convert PEM Certificate(s) to PFX `openssl pkcs12 -in CERTS.pem -nokeys -export -out CERTS.pfx` To include a key in PFX file use `-inkey KEY.pem` instead of `-nokeys`

PFX --> PEM
To extract everything within a PFX file as a PEM file: `openssl pkcs12 -in FILE.pfx -out EVERYTHING.pem -nodes`
To extract only the Private Key from a PFX file as PEM: `openssl pkcs12 -in FILE.pfx -out KEY.pem -nodes -nocerts`
PFX files can contain Certificate(s), or Certificate(s) + one matching Key `-clcerts` - extract only end-entity certificate (client certificate) `-cacerts` - extract all but end-entity certificate `-nokeys` - extract only certificates

OpenSSL Cheat Sheet - © Practical Networking .net pracnet.net/openssl

Enroll in Course

Course Summary

OpenSSL is the universal tool for **inspecting**, **diagnosing**, and **troubleshooting SSL & TLS**.

OpenSSL is composed of many different utilities, each of which is responsible for a specific aspect of the SSL and TLS ecosystem.

In this course, you will receiving training on how to use the following OpenSSL Utilities:

- openssl rsa
- openssl genrsa
- openssl dsa
- openssl gendsa
- openssl dsaparam
- openssl ec
- openssl ecparam
- openssl pkey
- openssl genpkey
- openssl x509
- openssl req
- openssl pkcs12

Each module contains demo files you can use to practice the commands along with each lesson.

The lessons in the course follow a section from the Cheat Sheet (available to download for free below).

After completing the course and downloading the OpenSSL Cheat Sheet,
you will be equipped to inspect and troubleshoot any SSL / TLS scenarios you find yourself in.

Course Curriculum

Welcome =)

1 Lesson

- Welcome =) START

Module 1 - Generating Public and Private Keys

4 Lessons

- Generating & Inspecting RSA Keys START
- Generating & Inspecting DSA Keys START
- Generating & Inspecting Elliptic Curve Keys START
- Module 1 Files START

Module 2 - Inspecting RSA, DSA, and EC Keys

4 Lessons

- Adding & Removing Encryption to RSA Keys START
- OpenSSL Pkey Utility START
- Matching Private Keys to Certificates and CSRs START

Module 2 Files START

Module 3 - Generating Certificates and CSRs

3 Lessons

- Creating a CSR and Certificate using an existing Private Key START
- Creating a CSR and Certificate using a new Private Key START
- Module 3 Files START

Module 4 - Inspecting Certificates and CSRs

2 Lessons

- Extracting specific information from Certificates and CSRs START
- Module 4 Files START

Module 5 - File Formats and Conversions (PEM, DER, PFX)

4 Lessons

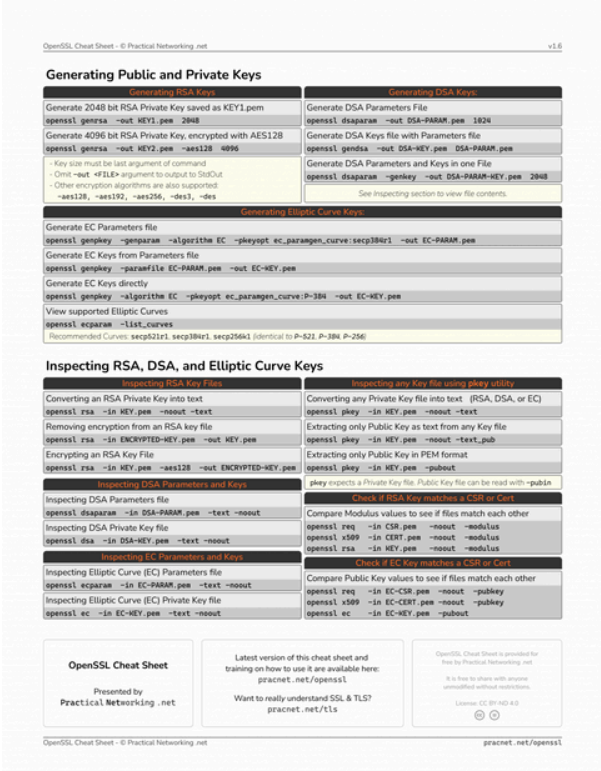
- Check if file is PEM, DER, or PFX START
- Converting files between PEM and DER formats START
- Converting files between PEM and PFX formats START
- Module 5 Files START

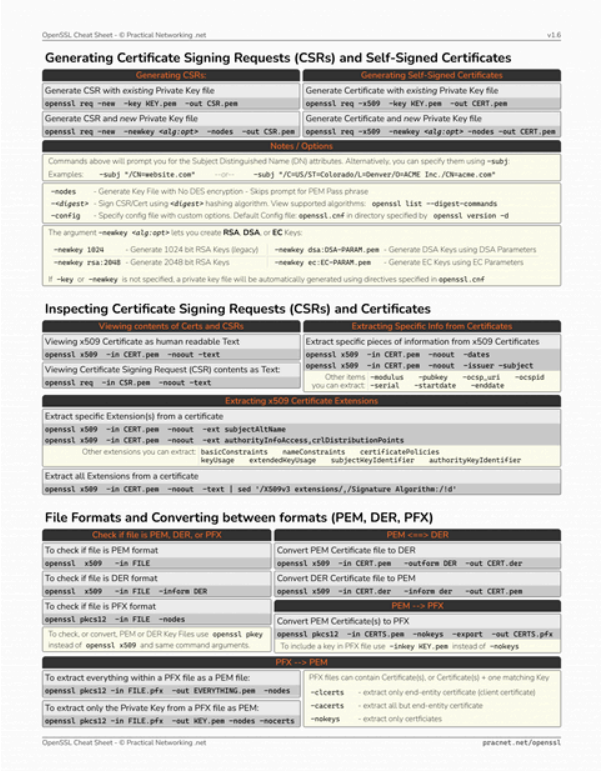
Course Pricing

OPENSSL TRAINING

$40 USD

Buy Now (/courses/openssl-training/buy/plan/61692)





Want to simply download the latest version of the OpenSSL Cheat Sheet (v1.7)?

Click Here (https://ln5.sync.com/dl/83b734df0/6pwcftu2-zvqpn8me-ba9evy2v-35xvawt8).