OpenSSL is an open-source command line tool that is commonly used to generate private keys, create CSRs, install your SSL/TLS certificate, and identify certificate information.

# OpenSSL and CSR Creation

The first step to obtaining an SSL certificate is using OpenSSL to create a certificate signing request (CSR) that can be sent to a Certificate Authority (CA)

The CSR contains the common name(s) you want your certificate to secure, information about your In order for a CSR to be created, it needs to have a private key from which the public key is extracted.

## Creating CSR

When generating a key, you have to decide three things: the key algorithm, the key size, and whether to use a passphrase.

make rsa with 2048 bit key : # **openssl genrsa -out *damain.pem*  2048**

decode the private key and view its contents: **# openssl rsa -text -in *damain.pem*  -noout**

NOTE :  **-noout** switch omits the output of the encoded version of the private key.

 pem file contains both the private key and the public key.

extract your public key from the key file: # **openssl rsa -in *damain.pem* -pubout -out public.key**

After generating your private key, you are ready to create your CSR. The CSR is created using the PEM format and contains the public key portion of the private key as well as information about you (or your company)

create a CSR using newly generated domain key: # **openssl req -new -key *domain.pem* -out *domain.csr***

After entering the command, you will be asked series of questions. Your answers to these questions will be embedded in the CSR. Answer the questions as described below:

|  |  |
| --- | --- |
| **Country Name (2 letter code)** | The two-letter country code where your company is legally located. |
| **State or Province Name (full name)** | The state/province where your company is legally located. |
| **Locality Name (e.g., city)** | The city where your company is legally located. |
| **Organization Name (e.g., company)** | Your company's legally registered name (e.g., YourCompany, Inc.). |
| **Organizational Unit Name (e.g., section)** | The name of your department within the organization. (You can leave this option blank; simply press **Enter**.) |
| **Common Name (e.g., server FQDN)** | The fully-qualified domain name (FQDN) (e.g., www.example.com). |
| **Email Address** | Your email address. (You can leave this option blank; simply press **Enter**.) |
| **A challenge password** | Leave this option blank (simply press **Enter**). |
| **An optional company name** | Leave this option blank (simply press **Enter**). |

some of the above CSR questions have default values that will be used if you leave the answer blank and press **Enter**

verify and  view the information in your CSR : **# openssl req -text -in domain.csr -noout -verify**

or to view csr file : **# cat domain.csr**

in this step .crt file will be created

# sending CSR to CA

By default, OpenSSL generates keys and CSRs using the PEM format. Sometimes it needs to convert to other format. The PKCS#12 format is an archival file that stores both the certificate and the private key. This format is useful for migrating certificates and keys from one system to another as it contains all the necessary files. PKCS#12 files use either the *.pfx* or *.p12* file extension.

view the contents of your certificate: **# openssl x509 -text -in yourdomain.crt -noout**

 convert PEM key and certificate into the PKCS#12 format: # **openssl pkcs12 -export -name "yourdomain” -out domain.pfx -inkey domain.pem -in domain.crt**

extract the pem from a PKCS#12 (.pfx) file and convert it into a PEM encoded private key:

**# openssl pkcs12 -in yourdomain.pfx -nocerts -out domain.pem -nodes**

extract the cert from a PKCS#12 (.pfx) file and convert it into a PEM encoded private key:

**# openssl pkcs12 -in yourdomain.pfx -nokeys -clcerts -out domain.crt**