**General OpenSSL Commands**

* **Generate a new private key and Certificate Signing Request**

openssl req -out CSR.csr -new -newkey rsa:2048 -nodes -keyout privateKey.key

* **Generate a self-signed certificate (see**[**How to Create and Install an Apache Self Signed Certificate**](https://www.sslshopper.com/article-how-to-create-and-install-an-apache-self-signed-certificate.html)**for more info)**

openssl req -x509 -sha256 -nodes -days 365 -newkey rsa:2048 -keyout privateKey.key -out certificate.crt

* **Generate a certificate signing request (CSR) for an existing private key**

openssl req -out CSR.csr -key privateKey.key -new

* **Generate a certificate signing request based on an existing certificate**

openssl x509 -x509toreq -in certificate.crt -out CSR.csr -signkey privateKey.key

* **Remove a passphrase from a private key**

openssl rsa -in privateKey.pem -out newPrivateKey.pem

## Checking Using OpenSSL

* **Check a Certificate Signing Request (CSR)**

openssl req -text -noout -verify -in CSR.csr

* **Check a private key**

openssl rsa -in privateKey.key -check

* **Check a certificate**

openssl x509 -in certificate.crt -text -noout

* **Check a PKCS#12 file (.pfx or .p12)**

openssl pkcs12 -info -in keyStore.p12

## Debugging Using OpenSSL

* **Check an MD5 hash of the public key to ensure that it matches with what is in a CSR or private 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

* **Check an SSL connection. All the certificates (including Intermediates) should be displayed**

openssl s\_client -connect www.paypal.com:443

## Converting Using OpenSSL

* **Convert a DER file (.crt .cer .der) to PEM**

openssl x509 -inform der -in certificate.cer -out certificate.pem

* **Convert a PEM file to DER**

openssl x509 -outform der -in certificate.pem -out certificate.der

* **Convert a PKCS#12 file (.pfx .p12) containing a private key and certificates to PEM**

openssl pkcs12 -in keyStore.pfx -out keyStore.pem -nodes

You can add -nocerts to only output the private key or add -nokeys to only output the certificates.

* **Convert a PEM certificate file and a private key to PKCS#12 (.pfx .p12)**

openssl pkcs12 -export -out certificate.pfx -inkey privateKey.key -in certificate.crt -certfile CACert.crt

### 1. **.crt (Certificate)**

* **تعریف**: فایل‌های .crt معمولاً شامل گواهی‌نامه‌های SSL/TLS هستند که توسط یک مرجع صدور گواهی (CA) صادر می‌شوند.
* **محتوا**: این فایل‌ها شامل کلید عمومی و اطلاعات مربوط به صاحب گواهی و CA هستند.
* **استفاده**: برای تأمین امنیت ارتباطات وب و احراز هویت سرورها استفاده می‌شوند.

**CA** یا **Certificate Authority** به معنای **مرجع صدور گواهی‌نامه** است. CA یک سازمان یا نهاد معتبر است که وظیفه‌اش صدور گواهی‌نامه‌های دیجیتال است.

### 2. **.csr (Certificate Signing Request)**

* **تعریف**: فایل .csr یک درخواست گواهی‌نامه است که معمولاً به یک CA ارسال می‌شود تا گواهی‌نامه صادر کند.
* **محتوا**: این فایل شامل اطلاعاتی درباره صاحب گواهی (مانند نام دامنه، سازمان، و کلید عمومی) است.
* **استفاده**: برای درخواست صدور گواهی‌نامه SSL/TLS از CA استفاده می‌شود.

### 3. **.pem (Privacy Enhanced Mail)**

* **تعریف**: فرمت .pem یک فرمت متنی است که برای ذخیره و انتقال اطلاعات امنیتی استفاده می‌شود.
* **محتوا**: این فایل‌ها می‌توانند شامل گواهی‌نامه‌ها، کلیدهای خصوصی، و اطلاعات دیگر باشند و معمولاً با برچسب‌های مشخصی محصور شده‌اند (مثل -----BEGIN CERTIFICATE-----).
* **استفاده**: در بسیاری از برنامه‌های کاربردی و سرورهای وب برای مدیریت گواهی‌نامه‌ها و کلیدها استفاده می‌شود.

### 4. **.pfx (Personal Information Exchange)**

* **تعریف**: فرمت .pfx یک فرمت باینری است که معمولاً برای ذخیره‌سازی گواهی‌نامه‌ها و کلیدهای خصوصی به کار می‌رود.
* **محتوا**: این فایل‌ها می‌توانند شامل کلیدهای عمومی و خصوصی و همچنین زنجیره گواهی‌نامه باشند.
* **استفاده**: معمولاً برای انتقال گواهی‌نامه‌ها و کلیدهای خصوصی بین سیستم‌ها و برای نصب در سرورهای ویندوز استفاده می‌شود.

### 5. **.p12 (PKCS#12)**

* **تعریف**: فرمت .p12 مشابه .pfx است و برای ذخیره‌سازی گواهی‌نامه‌ها و کلیدهای خصوصی استفاده می‌شود.
* **محتوا**: این فرمت شامل کلیدهای خصوصی و عمومی و زنجیره گواهی‌نامه‌ها است.
* **استفاده**: به‌طور معمول در سیستم‌های غیر ویندوزی و برای تأمین امنیت ارتباطات استفاده می‌شود.

### 6. **.key**

* **تعریف**: فایل‌های .key معمولاً شامل کلیدهای خصوصی هستند.
* **محتوا**: این فایل‌ها می‌توانند کلیدهای خصوصی برای استفاده در سرورهای وب یا سایر برنامه‌های امنیتی باشند.
* **استفاده**: برای تأمین امنیت ارتباطات و امضای دیجیتال استفاده می‌شوند.

### 7. **.der (Distinguished Encoding Rules)**

* **تعریف**: فرمت .der یک فرمت باینری است که برای ذخیره‌سازی گواهی‌نامه‌ها و کلیدها استفاده می‌شود.
* **محتوا**: این فایل‌ها معمولاً شامل گواهی‌نامه‌های X.509 هستند.
* **استفاده**: معمولاً در سیستم‌های غیر ویندوزی و برای اپلیکیشن‌هایی که نیاز به فرمت باینری دارند، استفاده می‌شود.

فرمت‌های **.pfx** و **.p12** هر دو برای ذخیره‌سازی گواهی‌نامه‌ها و کلیدهای خصوصی استفاده می‌شوند و به نوعی مشابه هستند، اما در زیر به تفاوت‌های اصلی آن‌ها اشاره می‌شود:

### 1. **توسعه‌دهنده و استاندارد**

* **.pfx**:
  + این فرمت معمولاً به عنوان یک استاندارد مایکروسافت شناخته می‌شود و بیشتر در سیستم‌عامل‌های ویندوز و برنامه‌های مایکروسافت استفاده می‌شود.
* **.p12**:
  + این فرمت بر اساس استاندارد PKCS#12 است که توسط RSA Security توسعه یافته و به‌طور گسترده در سیستم‌های مختلف و نرم‌افزارهای متنوع استفاده می‌شود.

### 2. **هماهنگی**

* **.pfx**:
  + به طور عمده در محیط‌های ویندوزی و نرم‌افزارهای مایکروسافت (مثل IIS) استفاده می‌شود.
* **.p12**:
  + در سیستم‌های غیر ویندوزی مانند لینوکس و مک نیز استفاده می‌شود و به‌طور کلی سازگاری بیشتری با نرم‌افزارها و پلتفرم‌های مختلف دارد.

### 3. **اسم‌گذاری**

* هر دو فرمت معمولاً به یک مفهوم اشاره دارند و در واقع می‌توانند به‌طور معمول به‌جای هم استفاده شوند. در بسیاری از موارد، یک فایل .pfx و یک فایل .p12 می‌توانند محتوای یکسانی داشته باشند.

### 4. **محتوا**

* هر دو فرمت می‌توانند شامل:
  + کلید خصوصی
  + کلید عمومی
  + زنجیره گواهی‌نامه‌ها (گواهی‌نامه‌های میانجی و ریشه)

به طور کلی، تفاوت‌های آن‌ها بیشتر به زمینه استفاده و سازگاری با سیستم‌های مختلف مربوط می‌شود تا محتوای واقعی آن‌ها. اگر سوال دیگری دارید یا به جزئیات بیشتری نیاز دارید، خوشحال می‌شوم کمک کنم!

### **Windows**

install Chocolatey

@"%SystemRoot%\System32\WindowsPowerShell\v1.0\powershell.exe" -NoProfile -InputFormat None -ExecutionPolicy Bypass -Command "iex ((New-Object System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))" && SET "PATH=%PATH%;%ALLUSERSPROFILE%\chocolatey\bin"

install OpenSSL

choco install openssl

### **Linux**

#### Ubuntu/Debian

sudo apt update

sudo apt install openssl

#### CentOS/RHEL

sudo yum install openssl

# ایجاد کلید خصوصی

openssl genrsa -out private.key 2048

# ایجاد CSR (Certificate Signing Request)

openssl req -new -key private.key -out request.csr

# ایجاد گواهی‌نامه خودامضا (Self-Signed Certificate)

openssl x509 -req -days 365 -in request.csr -signkey private.key -out certificate.crt

openssl pkcs12 -export -out certificate.pfx -inkey private.key -in certificate.crt

openssl genrsa -out private.key 2048

openssl req -new -key private.key -out request.csr -subj "/C=IR/ST=Tehran/L=Tehran/O=MCAC/CN=mcac.com"

openssl x509 -req -days 365 -in request.csr -signkey private.key -out certificate.crt

openssl pkcs12 -export -out certificate.pfx -inkey private.key -in certificate.crt -passout pass:Aa123456

# ایجاد کلید خصوصی با رمز عبور

openssl genrsa -aes256 -out private.key 2048



