PKI :

**A message encrypted by PrKey can only be decrypted by PuKey.**

(for data integrity => authentication : I possess the PriKey of the PubKey )

**A message encrypted by PuKey can only be decrypted by PrKey.**

(for confidentiality)

It is nearly impossible to guess PriKey from PubKey, but it is easy to generate

**Digital Signature :**

It is [Msg + PrKeyEn(Msg)], receiver can use PuKey to decrypt PrKeyEn(Msg) and compare with Msg to make sure that the integrity of message. To prove that message comes from PrKey owner.

**SSL Certificate :**

1. Contains PuKey, Certificate name, owner of public key, Digital Signature
2. Issued by a trusted CA ( HKPost)
3. Single Domain, Multi-Domain, Wildcard (\*.abc.com), Multi-wildcard-domain (\*.abc1.com . \*.api.abc2.com ….)
4. SSL Cert proves that you are communicating with that “person/server/domain”. CA signed (gives the digital signature )the cert to prove that what the cert describing is right.
5. If mutual authentication is needed.

**Steps to Apply SSL Cert :**

1. Generate keypair , generate CSR ( CSR contains Domain names, and other info, with public key ).
2. Send CSR to CA , SSL certificate will be returned

Useful Commands and scenarios :

Generate Keypair and CSR

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

server.key is the private key ,server.csr is the CSR to be sent to CA for signing

HKPost case :

openssl genrsa –des3 –out myserver.key 2048

openssl req –new –key myserver.key –out myserver.csr

HK POST return cer files and other root certs

P12

* (.p12 or pfx files contains both private key and Domain Cert), it is a common way to store PrKey and SSL Cert, and it is password protected.
* Programmer can import my certs and keys as keystore in for remote communication(allow remote server to trust ). (keystore are list of my cert to present to remote servers, and keystore is which remote cert I will trust)

Extract PriKey from p12

openssl pkcs12 -info -**in** INFILE.p12 -nodes -nocerts

Extract PriKey from p12

**openssl pkcs12 -in [yourfile.pfx] -clcerts -nokeys -out [mycert.crt]**

Apache Settings :

SSLCertificateFile "C:/SSL/server.cer" # SSL Certificate file

SSLCertificateKeyFile "C:/SSL/server.key"# private key file

SSLCertificateChainFile "C:/SSL/chain.cer"

A Scenario for server-to-server communication.

Extension things :

P12 – explained above

pem – just mean ----xxxx--- and base64 encoded, can be private key , can be cert file, apache support PEM format

cer/der – can be cert , can be private key , but in binary format, can be converted to PEM.

p7b – cert chain.

Cert Chain :

[Get your certificate chain right. As many know, certificates are not… | by Sebastiaan van Steenis | Medium](https://medium.com/@superseb/get-your-certificate-chain-right-4b117a9c0fce)