**Steps to create HTTPS :**

We use HTTS to secure communication between server and client.

As HTTP will transfer data as plain text.

HTTPS is protocol to encrypt request and response

**What is SSL ; secure socket layer**

It is standard security technology for establishing encrypted link between a web server and a

**Difference between SSL.2 and SSL.3**

SSL 3.0 uses HMAC, which is more powerful than MAC.

HMAC is 128-bit of encryption but MAC is 40 bit

MAC supports only RSA key exchange and not none-RSA

SSL 3 supports also chain certificate

1. **First we create CA certificate**

CA-Certificate is an entity that issues X509 ( standard entity trusted third party

openssl req -config c:\openssl\bin\openssl.cnf -new -x509 -keyout ca-key.pem.txt -out ca-certificate.pem.txt -days 365

**2. Creating the Keystore ( -genkey )**

A Java **KeyStore** (JKS) is a repository of security certificates – either authorization certificates or public key certificates – plus corresponding private keys, SSL encryption

The keys that server needs to use for secure communication are stored in a file called Keystore

keytool –keystore clientkeystore –genkey –alias client

Clinetkeystore file will be gerenarted

KeyTool will ask to enter password and company info

1. **Create CSR certificate request ( -certreq)**This file will be used by certificate to generate SLL that is presented to other parties during the handshake

keytool –keystore clientkeystore –certreq –alias client –keyalg rsa

–file client.csr

1. **Create CA-Key for CSR ( - CAkey)**

Openssl x509 -req -CA ca-certificate.pem.txt -CAkey ca-key.pem.txt

-in client.csr -out client.cer -days 365 -CAcreateserial

1. **Import CA certificate in to keystore file**

keytool -import -keystore clientkeystore -file ca-certificate.pem.txt

-alias theCARoot