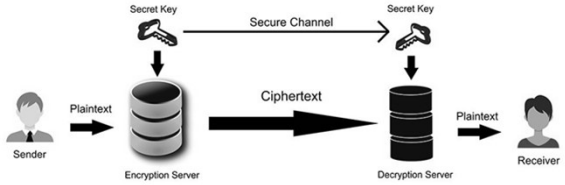
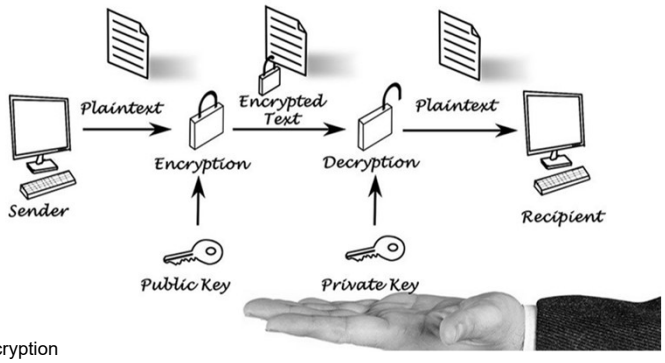
Symmetric Keys Distribution for End to End Encryption

William Stallings, Cryptography and Network Security



Symmetric Cryptography



https://www.trentonsystems.com/blog/symmetric-vs-asymmetric-encryption

Dr. Gaurav Varshney, IIT Jammu

Symmetric Keys Distribution for End to End Encryption

William Stallings, Cryptography and Network Security

1. A can select a key and physically deliver it to B.
2. A third party can select the key and physically deliver it to A and B.
3. If A and B have previously and recently used a key, one party can transmit the new key to the other, encrypted using the old key.
4. If A and B each has an encrypted connection to a third party C, C can deliver a key on the encrypted links to A and B.

1. The scale of the problem depends on the number of communicating pairs that must be supported.
2. If end-to-end encryption is done at a network or IP level, then a key is needed for each pair of hosts on the network that wish to communicate.
3. if there are hosts, the number of required keys is N(N-1)/2.
4. If encryption is done at the application level, then a key is needed for every pair of users or processes that require communication.

Symmetric Keys Distribution for End to End Encryption

William Stallings, Cryptography and Network Security

Centralized Key Distribution Center: Master Key Pre-shared Between all hosts and KDC

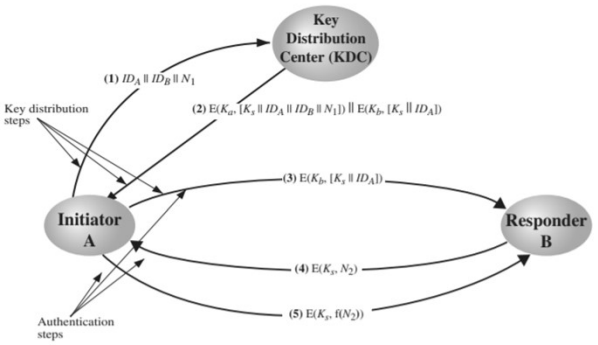


Figure 14.3 Key Distribution Scenario

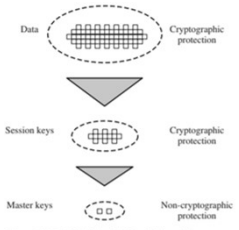


Figure 14.2 The Use of a Key Hierarchy

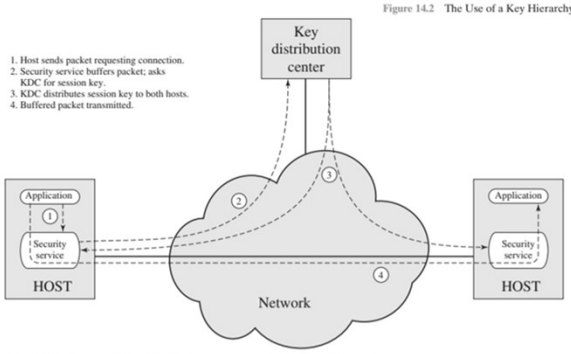


Figure 14.4 Automatic Key Distribution for Connection-Oriented Protocol

Symmetric Keys Distribution for End to End Encryption

William Stallings, Cryptography and Network Security

Decentralized Key Distribution

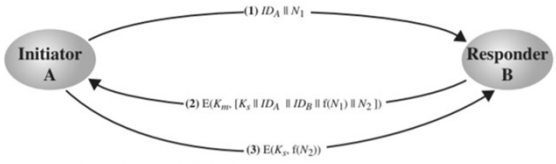


Figure 14.5 Decentralized Key Distribution

Key Distribution using Asymmetric Encryption

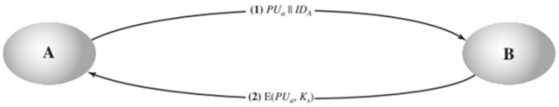


Figure 14.7 Simple Use of Public-Key Encryption to Establish a Session Key

A typical Man In The Middle Scenario

1. A generates a public/private key pair $\{PU_a, PR_a\}$ and transmits a message intended for B consisting of PU_a and an identifier of A, ID_A.
2. E intercepts the message, creates its own public/private key pair $\{PU_e, PR_e\}$ and transmits $PU_e \| ID_A$ to B.
3. B generates a secret key, K_s, and transmits $E(PU_e, K_s)$.
4. E intercepts the message and learns K_s by computing $D(PR_e, E(PU_e, K_s))$.
5. E transmits $E(PU_a, K_s)$ to A.

Symmetric Keys Distribution for End to End Encryption

William Stallings, Cryptography and Network Security

Decentralized Key Distribution

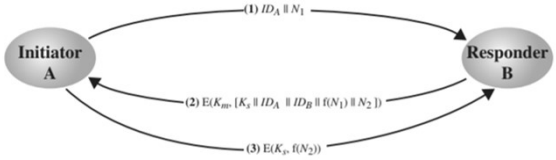


Figure 14.5 Decentralized Key Distribution

Key Distribution using Asymmetric Encryption

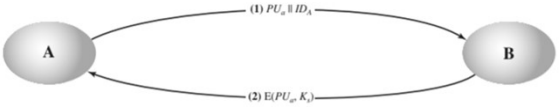


Figure 14.7 Simple Use of Public-Key Encryption to Establish a Session Key

A typical Man In The Middle Scenario

1. A generates a public/private key pair $\{PU_a, PR_a\}$ and transmits a message intended for B consisting of PU_a and an identifier of A, ID_A.
2. E intercepts the message, creates its own public/private key pair $\{PU_e, PR_e\}$ and transmits $PU_e \| ID_A$ to B.
3. B generates a secret key, K_s, and transmits $E(PU_e, K_s)$.
4. E intercepts the message and learns K_s by computing $D(PR_e, E(PU_e, K_s))$.
5. E transmits $E(PU_a, K_s)$ to A.

Symmetric Keys Distribution for End to End Encryption

William Stallings, Cryptography and Network Security

Key Distribution using Asymmetric Encryption with Confidentiality and Authentication

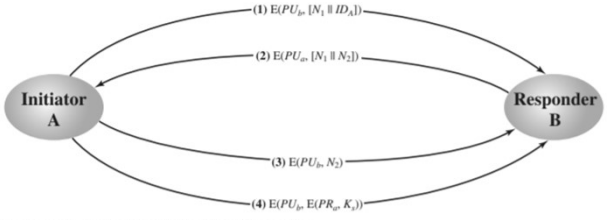


Figure 14.8 Public-Key Distribution of Secret Keys

Distribution of Public Keys

- Public announcement
- Publicly available directory
- Public-key authority
- Public-key certificates

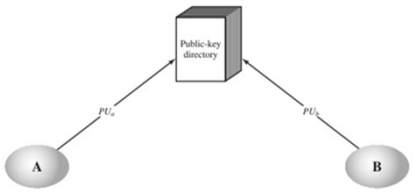


Figure 14.10 Public-Key Publication

shney, IIT Jammu

Symmetric Keys Distribution for End to End Encryption

William Stallings, Cryptography and Network Security

Public Key Authority

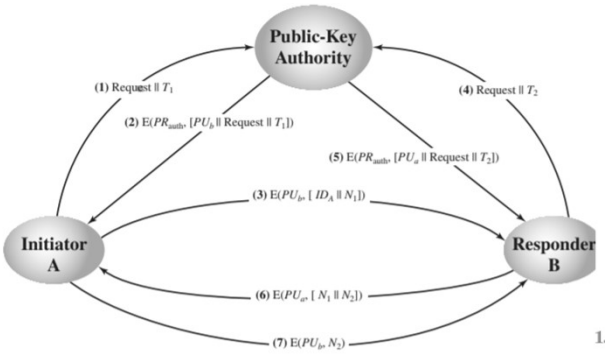


Figure 14.11 Public-Key Distribution Scenario

Public Key Certificates

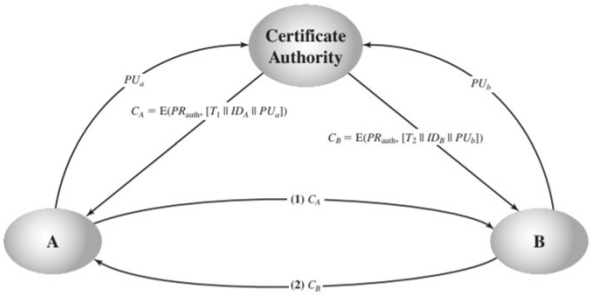


Figure 14.12 Exchange of Public-Key Certificates

1. Any participant can read a certificate to determine the name and public key of the certificate's owner.
2. Any participant can verify that the certificate originated from the certificate authority and is not counterfeit.
3. Only the certificate authority can create and update certificates.

Dr. Gaurav Varshney, IIT Jammu

End to End Encryption: Public Key Certificates

William Stallings, Cryptography and Network Security

Public Key Certificates Usage

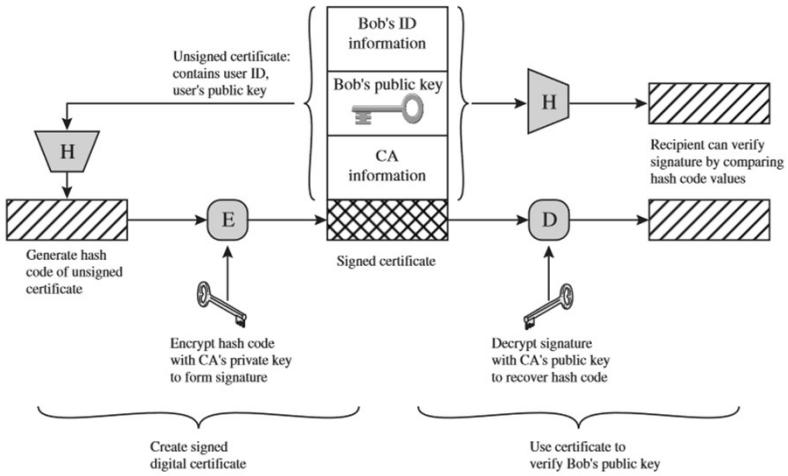


Figure 14.13 Public-Key Certificate Use

Public Key Infrastructure X.509-Architectural Model

William Stallings, Cryptography and Network Security

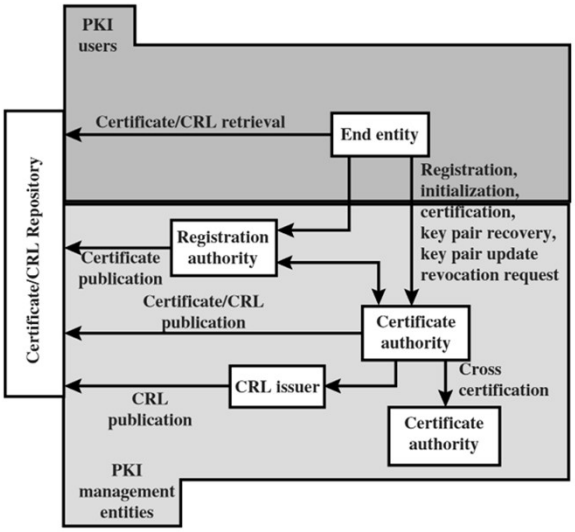


Figure 14.16 PKIX Architectural Model

acunetix

HTTPS: HTTP over TLS/SSL: History of TLS/SSL

- The Secure Sockets Layer (SSL) protocol was first introduced by Netscape in 1994.
- The first official release of SSL, version 2.0, was out in 1995
- The Transport Layer Security (TLS) protocol was first introduced in 1999 as an upgrade to SSL v3.
- TLS 1.1 April 2006, TLS 1.2 August 2008, TLS 1.3 August 2018

HTTPS: HTTP Secure, HTTP over TLS/SSL, Web PKI

- Web browsers know how to trust HTTPS websites based on certificate authorities that come pre-installed in their software.
- Certificate authorities (such as Let's Encrypt, DigiCert, Comodo, GoDaddy and Global Sign) are trusted by web browser.
- The security of HTTPS is that of the underlying TLS, which typically uses long-term public and private keys to generate a short-term session key, which is then used to encrypt the data flow between client and server. X.509 certificates are used to authenticate the server (and sometimes the client as well).

> When an end user accesses a website that has an HTTPS URL, they're interacting with Web PKI. It is a system of everything needed to issue, distribute and verify cryptographic keys and certificates, and tie them to the right domain. At the core of the Web PKI are cryptographic keys that enable cryptographic operations like authentication, authorisation and encryption. A certificate is, essentially, a binding of a cryptographic key (in this case a public key) to a web domain by a Certificate Authority (CA).

Dr. Gaurav Varshney, IIT Jammu

HTTPS: HTTP Secure, HTTP over TLS/SSL

- An X.509 [a standard that defines the format of public key certificates] certificate contains a public key and an identity (a hostname, or an organization, or an individual), and is either signed by a certificate authority or self-signed.
- When someone is holding a signed certificate from a trusted entity one can be sure of
  - That the public key in the certificate corresponds to the entity mentioned in the certificate or validate a document sent by the other party for authentication.
- X.509 also specifies the certificate revocation lists through which certificates which are no longed valid are announced or distributed.

X.509 Certificates

- There are different formats of X.509 certificates such as PEM, DER, PKCS#7 and PKCS#12.
- PEM[Privacy Enhanced Mail] and PKCS#7 [Public Key Cryptography Standards] formats use Base64 ASCII encoding while DER[Distinguished Encoding Rules] and PKCS#12 use binary encoding.
- The .pem file can include the server certificate, the intermediate certificate and the private key in a single file.
- The server certificate and intermediate certificate can also be in a separate .crt or .cer file.
- The PKCS#7 certificate uses Base64 ASCII encoding with file extension .p7b or .p7c

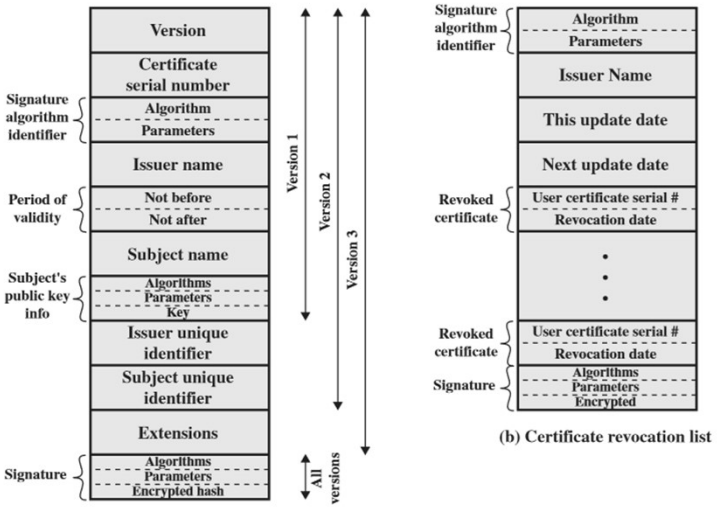


Figure 14.14 X.509 Formats

X.509 V3 Certificates: RFC 5280 format

```
Certificate  ::=  SEQUENCE  {
     tbsCertificate       TBSCertificate,
     signatureAlgorithm   AlgorithmIdentifier,
     signatureValue       BIT STRING  }

TBSCertificate  ::=  SEQUENCE  {
     version         [0]  EXPLICIT Version DEFAULT v1,
     serialNumber         CertificateSerialNumber,
     signature            AlgorithmIdentifier,
     issuer               Name,
     validity             Validity,
     subject              Name,
     subjectPublicKeyInfo SubjectPublicKeyInfo,
     issuerUniqueID  [1]  IMPLICIT UniqueIdentifier OPTIONAL,
                          -- If present, version MUST be v2 or v3
     subjectUniqueID [2]  IMPLICIT UniqueIdentifier OPTIONAL,
                          -- If present, version MUST be v2 or v3
     extensions      [3]  EXPLICIT Extensions OPTIONAL
                          -- If present, version MUST be v3
     }
```

X.509 V3

```
rtificateList  ::=  SEQUENCE  {
   tbsCertList          TBSCertList,
   signatureAlgorithm   AlgorithmIdentifier,
   signatureValue       BIT STRING  }

SCertList  ::=  SEQUENCE  {
   version                 Version OPTIONAL,
                                -- if present, MUST be v2
   signature               AlgorithmIdentifier,
   issuer                  Name,
   thisUpdate              Time,
   nextUpdate              Time OPTIONAL,
   revokedCertificates     SEQUENCE OF SEQUENCE  {
        userCertificate         CertificateSerialNumber,
        revocationDate          Time,
        crlEntryExtensions      Extensions OPTIONAL
                                 -- if present, version MUST be v2
                              }  OPTIONAL,
   crlExtensions           [0]  EXPLICIT Extensions OPTIONAL
                                 -- if present, version MUST be v2
                              }
```

CRL V2

Dr. Gaurav Varshney, IIT Jammu

PEM Format

```
-----BEGIN CERTIFICATE-----
MIIH/TCCBeWgAwIBAgIQaBYE3/M08XHYCnNVmcFBcjANBgkqhkiG9w0BAQsFADBy
MQswCQYDVQQGEwJVUzEOMAwGA1UECAwFVGV4YXMxEDAOBgNVBAcMB0hvdXN0b24x
ETAPBgNVBAoMCFNTTCBDb3JwMS4wLAYDVQQDDCVTU0wuY29tIEVWIFNTTCBJbnRl
cm1tZGlhdGUgQ0EgUlNBIFIzMB4XDTIwMDQwMTAwNTgzM1oXDTIxMDcxNjAwNTgz
M1owgb0xCzAJBgNVBAYTAlVTMQ4wDAYDVQQIDAVUZXhhczEQMA4GA1UEBwwHSG91
c3RvbjERMA8GA1UECgwIU1NMIENvcnAxFjAUBgNVBAUTDU5WMjAwODE2MTQyNDMx
FDASBgNVBAMMC3d3dy5zc2wuY29tMR0wGwYDVQQPDBRQcml2YXRlIE9yZ2FuaXph
dGlvbjEXMBUGCysGAQQBgjc8AgECDAZOZXZhZGExEzARBgsrBgEEAYI3PAIBAxMC
VVMwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDHheRkbb1FCc7xRKst
wK0JlGaKY8t7JbS2bQ2b6YUDgnHuIYHqBrCUV79oelikkokRkFvcvpaKinFHDQH
UpWEl6RUERYmSCg3O8Wi42uOcV2B5ZabmXCkwdxY5Ecl51BbM8UnGdoAGbdNmiRm
SmTjcs+hMxg4fFY6lBpiEVFiGUjGRR+61R67Lz6U4KleLNcCm07QwFYKBmpI08g
dyqSvRdUw55JopredjvVQtjUkB4hFT4GQX/ghf69Rkpz/+8u0dEQkhuUuucrqaIm
SGy43HRwBfDKFwYeWM7CPMd5e/dO+t08I8PbjzVTTv5hQDCsEYfV2T7AFf9ScNxM
kh7/AgMBAAGjggNBMIIDPTAfBgNVHSMEGDAWgBS/wVqH/yj6QT39t0/kHa+gYVgp
vTB/BggrBgEFBQcBAQRzMHEwTQYIKwYBBQUHMAKGQWh0dHA6Ly93d3cuc3NsLmNv
bS9yZXBvc2l0b3J5L1NTTGNvbST1dWJDQS1FVi1TU0wtUINBLTQwOTYtUjMuY3J0
MCAGCCsGAQUFBzABhhRodHRwOi8vb2NzcHMuc3NsLmNvbTAfBgNVHREEGDAWggt3
d3cuc3NsLmNvbYIHc3NsLmNvbTBfBgNVHSAEWDBWMAcGBWeBDAEBMA0GCyqEaAGG
9ncCBQEBMDwGDCsGAQQBgqkwAQMBBDAsMCoGCCsGAQUFBwIBFh5odHRwczovL3d3
dy5zc2wuY29tL3JlcG9zaXRvcnkwHQYDVR0lBBYwFAYIKwYBBQUHAwIGCCsGAQUF
BwMBMEgGA1UdHwRBMD8wPaA7oDmGN2h0dHA6Ly9jcmxzLnNzbC5jb20vU1NMY29t
LVN1YkNBLUVWLVNTTC1SU0EtND A5Ni1SMy5jcmwwHQYDVR0OBBYEFADAFUlazw5r
ZlHapnRxlUnpw+GLMA4GA1UdDwEB/wQEAwIFoDCCAX0GCisGAQQB1nkCBAIEggFt
BIIBaQFnAHcA9lyUL9F3MCIUVBgIMJRWjuNNExkzv98MLyALzE7xZOMAAAFxM0ho
bwAABAMASDBGAiEA6xeIlNR8Gk/63pYdnSvOx/CjptEMEv89iWWh1/urWlEClQDy
BreHU25DzwukQaRQjwV655ZLkqCnxbxQWRlOemj9JAB1AJQgvB6O1Y1siHMfgosi
LA3R2k1ebE+UPWHbTl9YTaLCAAABcTNlaNwAAAQDAEYwRAIgGRE4wzabNRdD8kq/
vFP3tQe2hm0x5nXulowh4lbw3lkCIFYb/3tSDplS7AcR4r+XpWtEKSTFWJmNCRbc
XJur2RGBAHUA7sCV7o1yZA+S48OSG8cSo2lqCXtLahoUOOZHssvtxfkAAAFxM0ho
8wAABAMARjBEAiB6IvboWss3R4ltVwjebl7D3yoFaX0NDh2dWhhgwCxrHwlgCfq7
ocMC5t+1ji5M5xaLmPC4I+WX3l/ARkWSyiO7lQcwDQYJKoZIhvcNAQELBQADggIB
ACeuur4QnujqmguSrHU3mhf+cJodzTQNqo4tde+PD1/eFdYAELu8xF+0At7xJiPY
i5RKwilyP56v+3iY2T9Iw7S8TJ041VLhalKp14MzSUzRyeoOAsJ7QADMCIHKUDlH
UU2pNuo88Y6igovT3bsnwJNiEQNqymSSYhktw0tadUoqjqXn06gsVioWTVDXysd5
qEx4t6slglcMm26YH1vJpCQEhKpc2y07gRkklBZRtMjThv4cXyyMX7uTcdT7AJBP
ueifCoVZ5JxXuo8d5139gwP1BAe7lBVPx2u7KN/UyOXdZmwMf/TmFGwDdCfsyHf/
ZsB2wLHozTYoAVmQ9FoU1JLgcVivqJ+vNlBhHXhbkMdN0j80R9Nz6ElglQjeK3O8
l/cFGm/B8+42hOlCld9ZdtndJcRJVjj0wD0qwevCafA9jJHv/jsE+t9Uz6cpCyh
sw+lrFdxUgqU58axqeK89FR+No4q0lIO+Ji1rJKr9nkSB0BqXozVnE1YB/KLvdls
uYZJuqb2pKku+zzTBgUvHUTZvBiNOtXL4hxvcJKT7WzOSd2wP10QI8DKg4vfiNDs
HWmB1c4KjBgOgA5uSUzaGmq/v4VncK5Ur+n9Lbfn fLc28J5ft/GotinMyDk3iar
F10YlqcOmeX1uFmKbdiXorGlkCoMF3TDx8rmp9DBiB/
-----END CERTIFICATE-----
```

DER Format

```
3082 07fd 3082 05e5 a003 0201 0202 1068
1604 dff3 34f1 71d8 0a73 5599 c141 7230
0d06 092a 8648 86f7 0d01 010b 0500 3072
310b 3009 0603 5504 0613 0255 5331 0e30
0c06 0355 0408 0c05 5465 7861 7331 1030
0e06 0355 0407 0c07 486f 7573 746f 6e31
1130 0f06 0355 040a 0c08 5353 4c20 436f
7270 312e 302c 0603 5504 030c 2553 534c
2e63 6f6d 2045 5620 5353 4c20 496e 7465
726d 6564 6961 7465 2043 4120 5253 4120
5233 301e 170d 3230 3034 3031 3030 3538
3333 5a17 0d32 3130 3731 3630 3035 3833
335a 3081 bd31 0b30 0906 0355 0406 1302
5553 310e 300c 0603 5504 080c 0554 6578
6173 3110 300e 0603 5504 070c 0748 6f75
7374 6f6e 3111 300f 0603 5504 0a0c 0853
534c 2043 6f72 7031 1630 1406 0355 0405
130d 4e56 3230 3038 3136 3134 3234 3331
1430 1206 0355 0403 0c0b 7777 772e 7373
6c2e 636f 6d31 1d30 1b06 0355 040f 0c14
5072 6976 6174 6520 4f72 6761 6e69 7a61
7469 6f6e 3117 3015 060b 2b06 0104 0182
373c 0201 020c 064e 6576 6164 6131 1330
1106 0b2b 0601 0401 8237 3c02 0103 1302
5553 3082 0122 300d 0609 2a86 4886 f70d
0101 0105 0003 8201 0f00 3082 010a 0282
0101 00c7 85e4 646d bd45 09ce f144 ab2d
c0ad 0920 668a 63cb 7b25 b4b6 6d0d 9be9
8209 0e09 c7b8 8607 a81a c251 5efd a1e9
6292 4a24 4641 6f72 fa5a 2a29 c51c 3407
5295 8423 a454 1116 2648 2837 3bc5 a2e3
6b8e 715d 81e5 969b 9970 a4c1 dc58 e447
25e7 505b 33c5 2719 da00 19b7 4d9a 2466
4a64 e372 cfa5 84cc 60e1 f158 ea50 6988
4545 8865 2319 147e eb54 7aec bcfa 5382
```

https://www.ssl.com/guide/pem-der-crt-and-cer-x-509-encodings-and-conversions/

Dr. Gaurav Varshney, IIT Jammu

HTTPS- X.509 Certificate Format

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            5c:2f:34:0f:87:ed:a3:d8:45:90:1e:7d
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C = BE, O = GlobalSign nv-sa, CN = AlphaSSL CA - SHA256 - G2
        Validity
            Not Before: May 11 10:02:42 2021 GMT
            Not After : Jun 12 10:02:42 2022 GMT
        Subject: CN = *.iitjammu.ac.in
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                RSA Public-Key: (2048 bit)
                Modulus:
                    00:c2:5c:9d:d2:f4:99:17:13:c6:78:fc:39:f8:70:
                    aa:bb:c8:f8:53:d0:9c:05:2b:7c:2d:d1:a2:a4:ea:
                    46:88:87:94:e8:17:bf:da:3c:b3:4a:7e:b9:54:72:
                    81:37:83:0b:22:be:e2:fe:98:c9:b2:6c:6e:24:c3:
                    0e:87:1b:d9:18:ba:2b:d9:07:fa:9f:f6:9a:d8:34:
                    04:07:26:23:e0:7e:4c:15:d0:7c:ee:7d:c9:38:8e:
                    75:bd:cc:10:e7:b7:7a:4b:d3:46:2e:02:1f:06:b3:
                    21:9e:4a:d1:fe:57:0e:ea:49:d1:cd:5b:9f:2e:c5:
                    b4:d0:f1:c9:8a:83:58:b3:03:4f:d8:52:6b:7c:b6:
                    ad:d0:57:24:ca:d8:98:10:fa:c8:04:e1:35:a8:04:
                    d4:90:0a:61:d5:35:d3:a8:90:83:75:5b:6d:c4:ad:
                    60:0b:92:8a:0b:74:e0:b0:15:cf:ba:95:27:5e:ba:
                    f1:8a:79:a8:a6:a9:47:77:37:f6:7e:f3:c1:c3:97:
                    ba:9b:2f:b2:f6:3f:73:8a:ba:ff:7e:04:08:f3:1e:
                    a4:38:bb:a9:c5:61:b1:2c:64:fb:2c:2c:3f:ba:e8:
                    e1:2f:01:83:f4:27:a4:6a:da:d0:04:a0:49:76:c1:
                    a8:85:ab:84:aa:57:1f:98:57:8b:8e:fc:a4:a4:1c:
                    61:91
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Key Usage: critical
                Digital Signature, Key Encipherment
            Authority Information Access:
                CA Issuers - URI:http://secure2.alphassl.com/cacert/gsalphasha2g2r1.crt
                OCSP - URI:http://ocsp2.globalsign.com/gsalphasha2g2

            X509v3 Certificate Policies:
                Policy: 1.3.6.1.4.1.4146.1.10.10
                  CPS: https://www.globalsign.com/repository/
                Policy: 2.23.140.1.2.1

            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 CRL Distribution Points:

                Full Name:
                  URI:http://crl2.alphassl.com/gs/gsalphasha2g2.crl

            X509v3 Subject Alternative Name:
                DNS:*.iitjammu.ac.in, DNS:iitjammu.ac.in
            X509v3 Extended Key Usage:
                TLS Web Server Authentication, TLS Web Client Authentication
            X509v3 Authority Key Identifier:
                keyid:F5:CD:D5:3C:08:50:F9:6A:4F:3A:B7:97:DA:56:83:E6:69:D2:68:F7

            X509v3 Subject Key Identifier:
                D8:42:12:45:DE:CB:F4:28:FE:BD:50:69:F3:B0:5F:89:24:49:EB:2B
            CT Precertificate SCTs:
                Signed Certificate Timestamp:
                    Version   : v1 (0x0)
                    Log ID    : 6F:53:76:AC:31:F0:31:19:D8:99:00:A4:51:15:FF:77:
                                15:1C:11:D9:02:C1:00:29:06:8D:B2:08:9A:37:D9:13
                    Timestamp : May 11 10:02:43.504 2021 GMT
                    Extensions: none
                    Signature : ecdsa-with-SHA256
                                30:45:02:20:6F:8D:F3:5E:72:CE:17:E2:CA:80:E7:15:
                                FF:16:4B:E3:14:49:7A:71:D9:88:2D:0C:C6:33:09:9A:
                                81:D1:DB:20:02:21:00:A5:7C:C1:FA:45:DD:55:52:73:
                                3A:F4:3F:52:C6:13:86:69:38:7F:59:DD:AC:0E:87:7E:
                                2E:5C:8E:34:F9:5D:FF
                Signed Certificate Timestamp:
                    Version   : v1 (0x0)
                    Log ID    : 29:79:BE:F0:9E:39:39:21:F0:56:73:9F:63:A5:77:E5:
                                BE:57:7D:9C:60:0A:F8:F9:4D:5D:26:5C:25:5D:C7:84
                    Timestamp : May 11 10:02:44.265 2021 GMT
                    Extensions: none
                    Signature : ecdsa-with-SHA256
                                30:44:02:20:3B:1A:E2:76:AA:5C:90:5A:A4:6A:6F:3E:
                                2F:3B:A8:C1:3A:16:0F:4E:0A:5E:C7:9F:35:6D:58:9A:
                                BD:6A:25:6F:02:20:23:B3:32:00:70:72:16:BB:4F:76:
                                09:31:24:9F:30:FA:1F:6A:C2:92:36:FD:49:67:3A:71:
                                44:99:39:1D:40:4E
                Signed Certificate Timestamp:
                    Version   : v1 (0x0)
                    Log ID    : 55:81:D4:C2:16:90:36:01:4A:EA:0B:9B:57:3C:53:F0:
                                C0:E4:38:78:70:25:08:17:2F:A3:AA:1D:07:13:D3:0C
                    Timestamp : May 11 10:02:43.545 2021 GMT
                    Extensions: none
                    Signature : ecdsa-with-SHA256
                                30:45:02:20:10:38:7C:DD:97:C7:CE:FB:5A:8B:F5:94:
                                B8:C0:FC:01:E8:1E:FE:DE:62:42:4C:FD:B7:29:1D:88:
                                AA:61:37:B4:02:21:00:A2:B2:E8:82:91:7A:2A:11:9B:
                                C4:F9:59:11:52:CA:0D:9C:51:29:66:B1:97:E3:6D:CF:
    Signature Algorithm: sha256WithRSAEncryption
         6d:c2:ec:ee:26:fc:cb:75:f8:7a:9f:be:d3:0d:24:8c:1e:f0:
         75:a5:3e:00:19:aa:b9:29:6d:92:06:c8:6e:55:5f:73:13:b8:
         c7:c6:b5:81:16:58:1e:8e:78:82:0b:0e:a8:74:f5:52:4b:e6:
         a3:3f:74:fc:57:f9:b9:07:01:b8:23:5c:ed:70:eb:b0:e2:0f:
         47:71:98:9f:a8:0a:62:bb:61:94:5e:5f:e6:6f:b0:c2:0a:54:
         29:a8:f8:8a:28:9e:f6:2b:40:7f:d4:41:cf:2d:79:58:ef:d5:
         9b:4e:12:80:6d:22:74:b9:1f:73:f6:37:d1af:e2:c7:bf:ca:
         69:d3:24:a6:c6:d2:75:53:c4:db:b3:81:1f:7d:86:fb:bc:7c:
         29:92:d9:ba:7c:ff:b7:2a:29:7b:79:46:10:03:2e:e7:ae:45:
```

HTTPS- X.509 Certificate Format

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            04:00:00:00:00:01:44:4e:f0:36:31
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C = BE, O = GlobalSign nv-sa, OU = Root CA, CN = GlobalSign Root CA
        Validity
            Not Before: Feb 20 10:00:00 2014 GMT
            Not After : Feb 20 10:00:00 2024 GMT
        Subject: C = BE, O = GlobalSign nv-sa, CN = AlphaSSL CA - SHA256 - G2
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                RSA Public-Key: (2048 bit)
                Modulus:
                    00:da:01:ec:e4:ec:73:60:fb:7e:8f:6a:b7:c6:17:
                    e3:92:64:32:d4:ac:00:d9:a2:0f:b9:ed:ee:6b:8a:
                    86:ca:92:67:d9:74:d7:5d:47:02:3c:8f:40:d6:9e:
                    6d:14:cd:c3:da:29:39:a7:0f:05:0a:68:a2:66:1a:
                    1e:c4:b2:8b:76:58:e5:ab:5d:1d:8f:40:b3:39:8b:
                    ef:1e:83:7d:22:d0:e3:a9:00:2e:ec:53:cf:62:19:
                    85:44:28:4c:c0:27:cb:7b:0e:ec:10:64:00:10:a4:
                    05:cc:a0:72:be:41:6c:31:5b:48:e4:b1:ec:b9:23:
                    eb:55:4d:d0:7d:62:4a:a5:b4:a5:a4:59:85:c5:25:
                    91:a6:fe:a6:09:9f:06:10:6d:8f:81:0c:64:40:5e:
                    73:00:9a:e0:2e:65:98:54:10:00:70:98:c8:e1:ed:
                    34:5f:d8:9c:c7:0d:c0:d6:23:59:45:fc:fe:55:7a:
                    86:ee:94:60:22:f1:ae:d1:e6:55:46:f6:99:c5:1b:
                    08:74:5f:ac:b0:64:84:8f:89:38:1c:a1:a7:90:21:
                    4f:02:6e:bd:e0:61:67:d4:f8:42:87:0f:0a:f7:c9:
                    04:6d:2a:a9:2f:ef:42:a5:df:dd:a3:53:db:98:1e:
                    81:f9:9a:72:7b:5a:de:4f:3e:7f:a2:58:a0:e2:17:
                    ad:67
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Key Usage: critical
                Certificate Sign, CRL Sign
            X509v3 Basic Constraints: critical
                CA:TRUE, pathlen:0
            X509v3 Subject Key Identifier:
                F5:CD:D5:3C:08:50:F9:6A:4F:3A:B7:97:DA:56:83:E6:69:D2:68:F7
            X509v3 Certificate Policies:
                Policy: X509v3 Any Policy
                  CPS: https://www.alphassl.com/repository/

            X509v3 CRL Distribution Points:

                Full Name:
                  URI:http://crl.globalsign.net/root.crl

                Full Name:
                  URI:http://crl.globalsign.net/root.crl

            Authority Information Access:
                OCSP - URI:http://ocsp.globalsign.com/rootr1

            X509v3 Authority Key Identifier:
                keyid:60:7B:66:1A:45:0D:97:CA:89:50:2F:7D:04:CD:34:A8:FF:FC:FD:4B

    Signature Algorithm: sha256WithRSAEncryption
         60:40:68:16:47:e7:16:8d:db:5c:a1:56:2a:cb:f4:5c:9b:b0:
         1e:a2:4b:f5:cb:02:3f:f8:0b:a1:f2:a7:42:d4:b7:4c:eb:e3:
         66:80:f3:25:43:78:2e:1b:17:56:07:52:18:cb:d1:a8:ec:e6:
         fb:73:3e:a4:62:8c:80:b4:d2:c5:12:73:a3:d3:fa:02:38:be:
         63:3d:84:b8:99:c1:f1:ba:f7:9f:c3:40:d1:58:18:53:c1:62:
         dd:af:18:42:7f:34:4e:c5:43:d5:71:b0:30:00:c7:e3:90:ae:
         3f:57:86:97:ce:ea:0c:12:8e:22:70:e3:66:a7:54:7f:2e:28:
         cb:d4:54:d0:b3:1e:62:67:08:f9:27:e1:cb:e3:66:b8:24:1b:
         89:6a:89:44:65:f2:d9:4c:d2:58:1c:8c:4e:c0:95:a1:d4:ef:
         67:2f:38:20:e8:2e:ff:96:51:f0:ba:d8:3d:92:70:47:65:1c:
         9e:73:72:b4:60:0c:5c:e2:d1:73:76:e0:af:4e:e2:e5:37:a5:
         45:2f:8a:23:3e:87:c7:30:e6:31:38:7c:f4:dd:52:ca:f3:53:
         04:25:57:56:66:94:e8:0b:ee:e6:03:14:4e:ee:fd:6d:94:64:
         9e:5e:ce:79:d4:b2:a6:cf:40:b1:44:a8:3e:87:19:5e:e9:f8:
         21:16:59:53
-----BEGIN CERTIFICATE-----
MIIETTCCAzWgAwIBAgILBAAAAAABRE7wNjEwDQYJKoZIhvcNAQELBQAwVzELMAkG
A1UEBhMCQkUxGTAXBgNVBAoTEEdsb2JhbFNpZ24gbnYtc2ExEDAOBgNVBAsTB1Jv
b3QgQ0ExGzAZBgNVBAMTEkdsb2JhbFNpZ24gUm9vdCBDQTAeFw0xNDAyMjAxMDAw
MDBaFw0yNDAyMjAxMDAwMDBaMEwxCzAJBgNVBAYTAkJFMRkwFwYDVQQKExBHbG9i
YWxTaWduIG52LXNhMSIwIAYDVQQDExlBbHBoYVNTTCBDQSAtIFNIQTI1NiAtIEcy
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA2gHs5OxzYPt+j2q3xhfj
kmQylKwA2aIPue3ua4qGypJn2XTXXUcCPI9A1p5tFM3D2ik5pw8FCmiiZhoexLKL
dljlq10dj0Cz0YvvHoN9ItDjqQAu7FPPYhmFRChMwCfLew7sEGQAEKQFzKByvkFs
MVtI5LHsu5PrVU3QfWJKpbSlpFmFxSWRpv6mCZ8GEG2PgQxkQF5zAJrgLmWYVBAA
cjJ14e00X9icxw3A1iNZRFz+VXqG7pRgIvGu0e7VRvaZxRs1dF+ssGSEj4k4HKGn
kCFPAm6946Fn1PhChw8K98kEbSqpL+9Cpd/do1PbmB6B+Zpye1reTz5/olg4het
ZwIDAQABo4IBIzCCAR8wDgYDVR0PAQH/BAQDAgEGMBIGA1UdEwEB/wQIMAYBAf8C
AQAwHQYDVR0OBBYEFPXN1TwIUPlqTzq3l9pWg+Zp0mj3MEUGA1UdIAQ+MDwwOgYE
VR0gADAyMDAGCCsGAQUFBwIBFiRodHRwczovL3d3dy5hbHBoYXNzbC5jb20vcmVw
b3NpdG9yeS8wMwYDVR0fBCwwKjAooCagJIYiaHR0cDovL2NybC5nbG9iYWxzaWdu
Lm5ldC9yb290LmNybDA9BggrBgEFBQcBAQQxMC8wLQYIKwYBBQUHMAGGIWh0dHA6
Ly9vY3NwLmdsb2JhbHNpZ24uY29tL3Jvb3RyMTAfBgNVHSMEGDAWgBRge2YaRQ2X
yolQL30EzTSo//z9SzANBgkqhkiG9w0BAQsFAAOCAQEAYEBoFkfnFo3bXKFWKsv0
XJuwWqJl9csCP/gLofKnOtS3TOvjZoDzJUN4LhsXVqdSGMvRqDzm+3M+pGKMgLTS
xRJzo9P6Aji+Yz2EuJnB8br3n8NA0VgYUBFi3a8YQn80TsVD1XGwMADH45CuP1eG
l87qDBKOInDjZqdUfy4oy9RU0LMeYmcI+Sfhy+NmuCQbiWqJRGXy2UzSWByMTsCV
odTvZy84IOgu/5ZR8LrYPZJwR2UcnnNytGAMXOLRc3bgr07i5TelRS+KIz6HxzDm
MTh89N1SyvNTBCVXVmaU6Avu5gMUTu79bZRknl7OedSyps9AsUSoPocZXun4IRZZ
Uw==
-----END CERTIFICATE-----
```


HTTPS

- Certificate Signing Request [CSR]
  - An organization looking for a public key certificate raise a CSR.
  - The organization generates a Private Key, Public Key Pair and provides the public key along with information that identifies domain name and organizational identity requesting a public key certificate.
  - The CA issues a certificate binding the public key with the distinguished name.
    - This is because CA hashes the contents of the certificate and encrypts it with its private key
      - The digital signature is appended to the end of the public key certificate issued to an organization
  - CA public key certificates are required for verification of the digital certificate issued by the CA to other entities.
  - CA certificate is trusted as it is signed in a similar manner by a root CA.
  - Root CA certificate is always trusted by default. Browsers are pre loaded with root CA certificates. Root CA certificate are signed by Root CA itself.

```
-----BEGIN CERTIFICATE REQUEST-----
MIIByjCCATMCAQAwgYkxCzAJBgNVBAYTAlVTMRMwEQYDVQQIEwpDYWxpZm9ybmlh
MRYwFAYDVQQHEw1Nb3VudGFpbiBWaWV3MRMwEQYDVQQKEwpHb29nbGUgSW5jMR8w
HQYDVQQLExZJbmZvcm1hdGlvbiBUZWNobm9sb2d5MRcwFQYDVQQDEw53d3cuZ29v
Z2xlLmNvbTCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEApZtYJCHJ4VpVXHfV
IlstQTlO4qC03hjX+ZkPyvdYd1Q4+qbAeTwXmCUKYHThVRd5aXSqlPzyIBwieMZr
WFlRQddZ1IzXAlVRDWwAo60KecqeAXnnUK+5fXoTI/UgWshre8tJ+x/TMHaQKR/J
cIWPhqaQhsJuzZbvAdGA80BLxdMCAwEAAaAAMA0GCSqGSIb3DQEBBQUAA4GBAIhl
4PvFq+e7ipARgI5ZM+GZx6mpCz44DTo0JkwfRDf+BtrsaC0q68eTf2XhYOsq4fkH
Q0uA0aVog3f5iJxCa3Hp5gxbJQ6zV6kJ0TEsuaaOhEko9sdpCoPOnR8m2i/XRD2D
6iNh8f8z0ShGsFqjDgFHyF3o+1Uyj+UC6H1QW7bn
-----END CERTIFICATE REQUEST-----
```

https://www.digicert.com/kb/ssl-support/openssl-quick-reference-guide.htm

HTTPS

```
Certificates (2976 bytes)
    Certificate Length: 1734
  Certificate: 308206c2308205aaa003020102020900ae3e09b992ffbd9b… (id-at-commonName=*.niituniversity.in,id-at-organizationalUnitName=Domain Control Validated)
    signedCertificate
         version: v3 (2)
         serialNumber: 12555483503795355035
      signature (sha256WithRSAEncryption)
           Algorithm Id: 1.2.840.113549.1.1.11 (sha256WithRSAEncryption)
      issuer: rdnSequence (0)
        rdnSequence: 6 items (id-at-commonName=Go Daddy Secure Certificate Authority - G2,id-at-organizationalUnitName=http://certs.godaddy.com/repositor,id-at-o
      validity
        notBefore: utcTime (0)
        notAfter: utcTime (0)
      subject: rdnSequence (0)
      subjectPublicKeyInfo
        algorithm (rsaEncryption)
        subjectPublicKey: 3082010a0282010100999674d297ced06fe745cf51a6e0613…
             modulus: 0x0099674d297ced06fe745cf51a6e0613d346e885b8754b11…
             publicExponent: 65537
      extensions: 10 items
    algorithmIdentifier (sha256WithRSAEncryption)
      Padding: 0
      encrypted: 6cb861ab5417a3aa72f54809a5e5686b95a093c21348d1e7…
    Certificate Length: 1236
  Certificate: 308204d0308203b8a003020102020107300d06092a864886… (id-at-commonName=Go Daddy Secure Certificate Authority - G2,id-at-organizationalUnitName=http://c
```


HTTPS

•A certificate chain is a list of certificates (usually starting with an end-entity certificate) followed by one or more CA certificates (usually the last one being a self-signed certificate), with the following properties:

- The Issuer of each certificate (except the last one) matches the Subject of the next certificate in the list.
- Each certificate (except the last one) is supposed to be signed by the secret key corresponding to the next certificate in the chain (i.e. the signature of one certificate can be verified using the public key contained in the following certificate).
- The last certificate in the list is a trust anchor: a certificate that you trust because it was delivered to you by some trustworthy procedure.

HTTPS-
Certificate Chain Validation

ney, IIT Jammu

Intermediate CA Snapshot

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            04:00:00:00:00:01:44:4e:f0:42:47
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=BE, O=GlobalSign nv-sa, OU=Root CA, CN=GlobalSign Root CA
        Validity
            Not Before: Feb 20 10:00:00 2014 GMT
            Not After : Feb 20 10:00:00 2024 GMT
        Subject: C=BE, O=GlobalSign nv-sa, CN=GlobalSign Organization Validation CA - SHA256 - G2
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
```

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            10:e6:fc:62:b7:41:8a:d5:00:5e:45:b6
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=BE, O=GlobalSign nv-sa, CN=GlobalSign Organization Validation CA - SHA256 - G2
        Validity
            Not Before: Nov 21 08:00:00 2016 GMT
            Not After : Nov 22 07:59:59 2017 GMT
        Subject: C=US, ST=California, L=San Francisco, O=Wikimedia Foundation, Inc., CN=*.wikipedia.org
        Subject Public Key Info:
            Public Key Algorithm: id-ecPublicKey
                Public-Key: (256 bit)
                pub:
                    04:c9:22:69:31:8a:d6:6c:ea:da:c3:7f:2c:ac:a5:
                    af:c0:02:ea:81:cb:65:b9:fd:0c:6d:46:5b:c9:1e:
                    ed:b2:ac:2a:1b:4a:ec:80:7b:e7:1a:51:e0:df:f7:
                    c7:4a:20:7b:91:4b:20:07:21:ce:cf:68:65:8c:c6:
                    9d:3b:ef:d5:c1
```

End Entity Certificate Snapshot

Root CA/Trust Anchor Certificate

```
Certificate:[14]
    Data:
        Version: 3 (0x2)
        Serial Number:
            04:00:00:00:00:01:15:4b:5a:c3:94
    Signature Algorithm: sha1WithRSAEncryption
        Issuer: C=BE, O=GlobalSign nv-sa, OU=Root CA, CN=GlobalSign Root CA
        Validity
            Not Before: Sep  1 12:00:00 1998 GMT
            Not After : Jan 28 12:00:00 2028 GMT
        Subject: C=BE, O=GlobalSign nv-sa, OU=Root CA, CN=GlobalSign Root CA
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:da:0e:e6:99:8d:ce:a3:e3:4f:8a:7e:fb:f1:8b:
```