Digital Certificates Overview

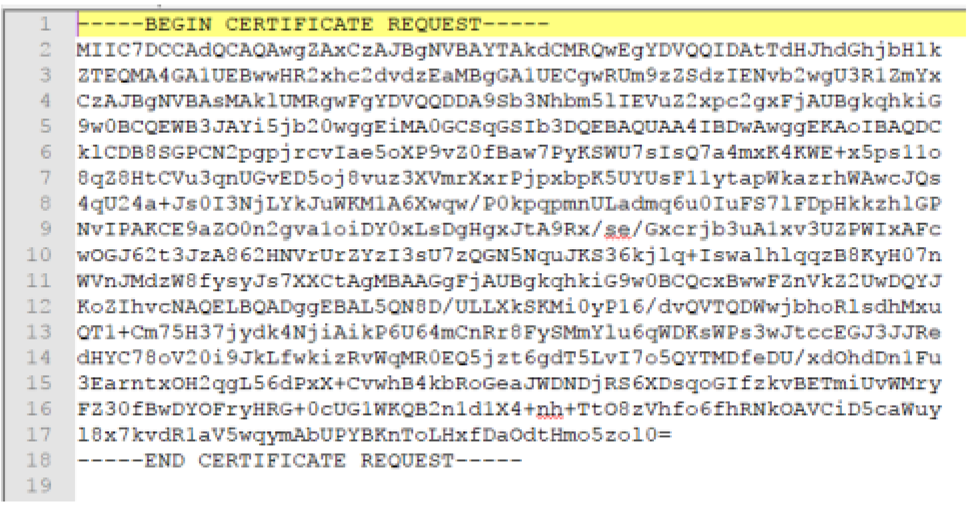
In order to understand the TLS protocol we need to understand a digital certificate. So far, we have worked on the basis that we can trust the key provided by a server, but what if they are not who they claim to be? If Eve intercepts the initial key exchange then the key of the server could be replaced. In order to authenticate the server to the client, we can make use of digital certificates.

A digital certificate is a collection of data which associates a public key with a server. A certificate authority is a trusted third party who verify a server belongs to the entity claiming it, and associates the entity’s public key with that server. The certificate authority digitally signs the public key and other information (i.e. the digital certificate) using their signing key to demonstrate its trust in the server’s identity.

An example certificate authority is Verisign and an example entity is Google. If google wanted to add a new server, then they could submit a certificate signing request to the certificate authority.

A certificate signing request contains a range of information. It's a block of text encoded often in ASN.1. ASN stands for abstract syntax notation. It is typically generated on a server where the certificate is going to be installed, and contains a public key; a common name, which is fully qualified domain name of the server; the organisational name; and the unit which is responsible for the server, such as the IT department. It also contains the city, the state, county or region, the country, and an email address of a person responsible for the server.

Figure 1 shows an example of what a CSR looks like when generated using OpenSSL. We can see that it's encoded, but when we open it in something like OpenSSL, it will display the full details.



```
-----BEGIN CERTIFICATE REQUEST-----
MIIC7DCCAdQCAQAwgZAxCzAJBgNVBAYTAkdCMRQwEgYDVQQIDAtTdHJhdGhjbHlk
ZTEQMA4GA1UEBwwHR2xhc2dvdzEaMBgGA1UECgwRUm9zZSdzIENvb2wgU3R1ZmYx
CzAJBgNVBAsMAklUMRgwFgYDVQQDDA9Sb3Nhbm51IEVuZ2xpc2gxFjAUBgkqhkiG
9w0BCQEWB3JAYi5jb20wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDC
k1CDB8SGPCN2pgpjrcvIae5oXP9vZ0fBaw7PyKSWU7sIsQ7a4mxK4KWE+x5ps1lo
8qZ8HtCVu3qnUGvED5oj8vuz3XVmrXxrPjpxbpK5UYUsFllytapWkazrhWAwcJQs
4qU24a+Js0I3NjLYkJuWKM1A6Xwqw/P0kpqpmnULadmq6u0IuFS7lFDpHkkzhlGP
NvIPAKCE9aZO0n2gva1oiDY0xLsDgHgxJtA9Rx/se/Gxcrjb3uA1xv3UZPWIxAFc
wOGJ62t3JzA862HNVrUrZYzI3sU7zQGN5NquJKS36kjlq+Iswa1hlqqzB8KyH07n
WVnJMdzW8fysyJs7XXCtAgMBAAGgFjAUBgkqhkiG9w0BCQcxBwwFZnVkZ2UwDQYJ
KoZIhvcNAQELBQADggEBAL5QN8D/ULLXkSKMi0yP16/dvQVTQDWwjbhoRlsdhMxu
QT1+Cm75H37jydk4NjiAikP6U64mCnRr8FySMmYlu6qWDKsWPs3wJtccEGJ3JJRe
dHYC78oV20i9JkLfwkizRvWqMR0EQ5jzt6gdT5LvI7o5QYTMDfeDU/xdOhdDn1Fu
3EarntxOH2qgL56dPxX+CvwhB4kbRoGeaJWDNDjRS6XDsqoGIfzkvBETmiUvWMry
FZ30fBwDYOFryHRG+0cUG1WKQB2n1d1X4+nh+TtO8zVhfo6fhRNkOAVCiD5caWuy
l8x7kvdR1aV5wqymAbUPYBKnToLHxfDaOdtHmo5zol0=
-----END CERTIFICATE REQUEST-----
```

Figure 1

After receiving such a request, the certificate authority must ensure the server and public key belongs to the entity claiming ownership. The certificate authority can complete a number of checks. One way of achieving this is by domain validation. So an email could be sent to an administration email responsible for that domain, and they can have something in there like an authentication token or a link. If that link is then used, they prove a level of access to that domain. They can also perform research to make sure that they're happy that that server does indeed belong to the company. Once satisfied, the certificate authority can then take the public key provided in the certificate signing request and associate it with that company and the certificate.

A digital certificate has a name, a time frame for which it is valid, who it was issued by, the owner's public key, and the certificate authority also digitally signs the certificate with their signing key. Effectively, they're vouching for the company.

But how does a client know that it is a valid certificate and from the certificate authority? Well, a number of certificates are pre-installed on a client's operating system, such as VeriSign. This means that they have the public key for VeriSign, meaning that we can check the digital signature on the digital certificate that's provided. Since this is in the operating system, the client can trust certificates from VeriSign that it can decrypt using VeriSign's verifying key.

When a client downloads the certificate from the web server it is trying to communicate with, the client can then verify the signature to see if it is correct and check the integrity of the certificate as well as the details such as ensuring it is not out of date. Web browsers often show when a digital certificate may not be trusted, e.g. if it is a self-signed certificate.