**HRM CODE TUTORIAL COLLECTION**

**Collection By:** R&D Team

Index

[REQUEST POST LARGE DATA JSON – WEB MVC 5](#_Toc23858538)

[Features 5](#_Toc23858539)

[Resolve Exception 5](#_Toc23858540)

[Access to the Code 5](#_Toc23858541)

[How to use 5](#_Toc23858542)

[Example 6](#_Toc23858543)

[Access to the Code 19](#_Toc23858558)

[How to use 19](#_Toc23858559)

[Example 20](#_Toc23858560)

# 

# SSL Certificate

## What is an SSL Certificate

### What is an SSL Certificate and How does it work

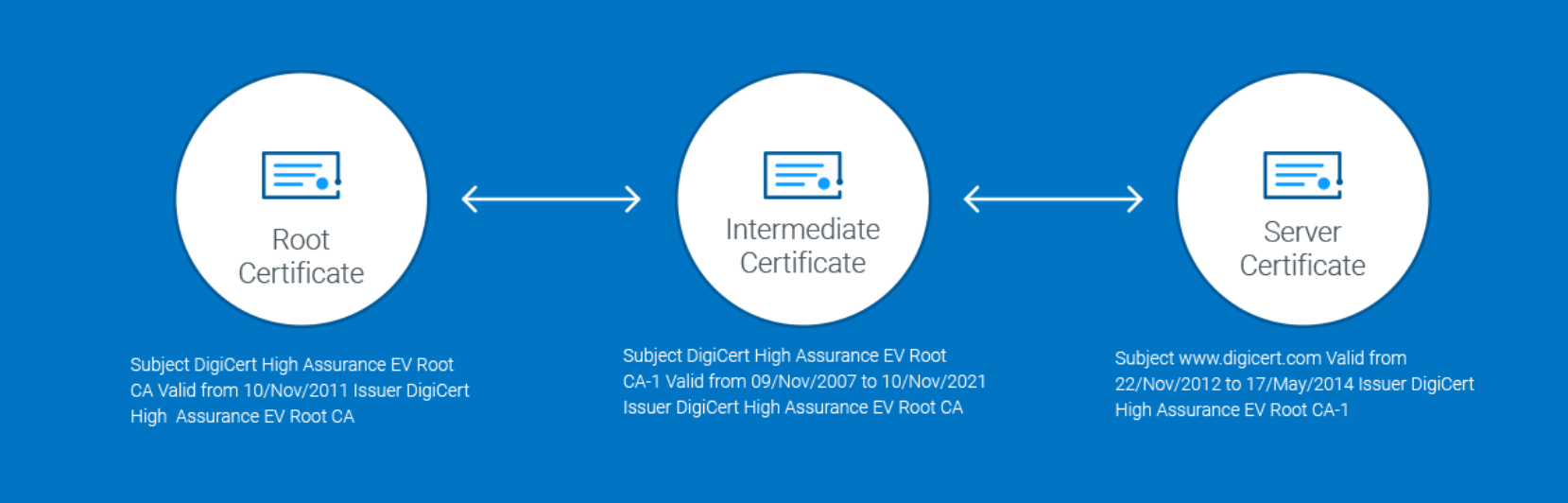
SSL certificate tạo ra một Encrypted Connection và Establish Trust.

Một trong những bộ phận quan trọng nhất của Online Business là tạo ra một Trusted Environment nơi mà Customer có thể tin tưởng để sử dụng. SSL Certificate tạo ra một Foundation of Trust bằng cách tạo nên một Secure Connection. Để đảm bảo các kết nối của người dùng là an toàn, Browsers cung cấp Visual Cues mà chúng ta gọi là EV Indicators.

SSL Certificate có một cặp Key – một Public và một Private. Cặp Keys hoạt động cùng với nhau để tạo nên một Encrypted Connection. Certificate cũng giữ một thứ gọi là “subject”, nó là định danh của Certificate/Website Owner.

Để nhận được một Certificate, chúng ta cần tạo ra một Certificate Signin Request (CSR) tại Server của chúng ta. Quá trình này tạo ra một Private Key và một Public Key trên Server. CSR Data File mà chúng ta gửi cho SSL Certificate Issuer (gọi là Certificate Authority hoặc CA) chứa Public Key. Và CA sử dụng CSR Data File để tạo ra một Data Structure match với Private Key của chúng ta mà không ảnh hưởng tới bản thân Key. CA sẽ không bao giờ thấy Private Key

Một khi chúng ta nhận được SSL Certificate, chúng ta cần cài đặt nó trên Server. Chúng ta cũng cần cài đặt một Intermediate Certificate để tạo nên một SSL Certificate đáng tin cậy bằng cách gắn nó vào Root Certificate của CA. Quá trình cài đặt và kiểm tra Certificate sẽ khác nhau phụ thuộc vào Server của chúng ta.



Như trong hình phía trên, chúng ta có thể thấy được thứ gợi là Certificate Chain, nó kết nối Server Certificate của chúng ta tới Root Certificate của CA (DigiCert) thông qua một Intermediate Certificarte.

Phần quan trọng nhất của SSL Certificate là nó là một Digitally Signed bởi một Trusted CA như DigiCert. Bất kỳ ai cũng có thể tạo một Certificate, nhưng Browser chỉ tin tưởng duy nhất những Certificates được tạo từ một tổ chức nằm trong danh sách Trusted CAs, được biết tới với tên là Trusted Root CA Store. Để có thể được thêm vào Trusted Root CA Store và trở thành một Certificate Authority, một công ty phải tuân thủ và được kiểm tra thông qua Security và Authentication Standard được Browsers thiết lập.

Một SSL Certificate được phát hành bởi một CA cho một tổ chức và Website/Domain của họ sẽ tiến hành xác thực rằng một Trusted Third Party có thể được chứng thực thông qua Identity của tổ chức đó. Bởi vì Browser tin tưởng CA, Browser sẽ tin tưởng luôn Organization’s Idenity. Browser sẽ cho User biết rằng Website có được bảo vệ, và User có thể cảm thấy an toàn khi sử dụng Website và đưa những thông tin cá nhân lên đó.

#### What is Secure Sockets Layer (SSL)

Security Sockets Layer (SSL) là một Standard Security Technology sử dụng để tạo ra một Encrypted Link giữa một Server và một Client – thông thường là một Web Server (Website) và một Browser, hoặc một Email Server và Mail Client (e.g., Outlock). Nó được biết tới nhiều hơn là Transport Layer Security (TLS) – một Successor Technology của SSL.

SSL cho phép các Sensitive Information như là Credit Card Numbers, Social Security Number hoặc Login Credentials có thể được gửi đi một cách an toàn. Thông thường, Data được gửi giữa các Browsers và Web Servers dưới dạng Plain Text – điều này khiến chúng ta có thể rơi vào trạng thái dễ bị tấn công. Nếu một Attacker có thể can thiệp vào tất cả Data được gửi giữa Browser và Web Server, chúng có thể xem được toàn bộ thông tin của chúng ta.

Nói một cách chuyên ngành hơn. SSL là một Security Protocol. Protocol mô tả cách mà các thuật toán nên được sử dụng. Trong trường hợp này, SSL Protocol chỉ định các Variables của Encryption cho cả Link và Data được gửi.

Tất cả Browsers có khả năng tương tác với Secured Web Server thông qua SSL Protocol. Tuy nhiên Browser và Server cần cái gọi là SSL Certificate để có thể tạo được một kết nối an toàn.

SSL-Secured Websites luôn luôn là HTTPS thay vì HTTP

#### How does the SSL Certificate create a new Secure Connection?

Khi một Browser có ý định truy cập tới một Website được bảo vệ bởi SSL, Browser và Web Server tạo nên một kết nối SSL sử dụng một quá trình gọi là “SSL Handshake”. Chú ý rằng SSL Handshake không thể được thấy từ người dùng và nó xảy ra ngay lập tức

Về cơ bản, có 3 Keys được sử dụng để Set up SSL Connection: Public Key, Private Key và Session Key. Tất cả mọi thứ được mã hóa với Public Key chỉ có thể giải mã bằng Private Key và ngược lại. Bởi vì quá trình mã hóa và giải mã với Private và Public Keys cần một khổi lượng lớn Processing Power, vì vậy nó chỉ được sử dụng duy nhất tại quá trình SSL Handshake để tạo ra một Symmetric Session Key (Session Key đối xứng). Sau khi Secure Connection được tạo, Session Key sẽ được sử dụng để mã hóa tất cả Transmited Data.



1. Browser kết nối tới một Web Server được bảo về bởi SSL (https). Browserr requests định danh của Server
2. Server gửi một bản Copy SSL Certificate của nó, bao gồm cả Public Key của Server
3. Browser kiểm trả Certificate Root danh sách Trusted CAs và Certificate có hết hạn, Unrevoked, và Common Name của nó có hợp lệ hay không để kết nối tới. Nếu Browser tin tưởng Certificate, nó sẽ tạo, mã hóa và gửi lại một Symmetric Session Key sử dụng Public Key của Server
4. Server giải mã Symmetric Session Key sử dụng Private Key và gửi một Acknowledgement Encrypted với Session Key để bắt đầu một Session được mã hóa
5. Server và Browser bây giờ đã mã hóa toàn bộ Data mà chúng truyền qua lại với Session Key

#### Is my Certificate SSL or TLS?

SSL Protocol luôn luôn được sử dụng để mã hóa và bảo vệ các Data được truyền đi. Mỗi lần một hoặc nhiều Secure Version được released, chỉ duy nhất Version Number được thay đổi để thể hiện sự thay đổi (e.g., SSLv2.0). Tuy nhiên, tới khi cần nâng cấp từ SSLv3.0 thay vì gọi New Version SSLv4.0, nó được đổi tên thành TLSv1.0, chúng ta hiện tạ ở TLSv1.3

Bởi vì SSL hiện vẫn được biết đến rộng rãi hơn, và được sử dụng phổ biến nên DigiCert sử dụng TLS/SSL khi chỉ tới việc Certificate hoặc mô tả bằng cách nào mà Data được bảo vệ. Khi chúng ta muc một SSL Certificate (e.g., từ Standard SSL, Extended Validation SSL. Etc.), chúng ta thực chất nhận được một TLS Certificate (RSA hoặc ECC)

## Creating a Self-Signed SSL Certificate – Using openssl tool

### What is a Self-Signed SSL Certificate?

Một Self-Signed SSL Certificate là một Certificate được đăng ký bởi người tạo ra nó thay vì một Trusted Certificate Authority. Self-Signed Certificate cót thể có cùng mức độ mã hóa so với Trusted CA-Signed SSL Certificate

Web Browsers không thể nhận ra việc Self-Signed Certificate là hợp lệ, nên khi sử dụng Self-Signed Certificate, Browser sẽ thông báo người truy cập là Website Certificate chưa được xác thực.

Thông thường, Self-Signed Certificate được sử dụng cho mục đích Test và Internal Usage. Chúng ta không nê sử dụng Self-Signed Certificate trong Production Systems đưa lên Internet.

### Prerequisites

OpenSSL Toolkit