SSL/TLS CERTIFICATES THE ULTIMATE GUIDE

AUTHOR:
ABOUTSSL.ORG

ABOUTSSL

EXECUTIVE SUMMARY

The eBook "SSL/TLS Certificates: The Ultimate Guide" presents guidance on SSL/TLS Certificates. The material in this eBook is easy to understand and it covers a comprehensive list of details for anyone whether it's a student, web hosting provider, or anyone else interested in learning about SSL security.

If you are someone with computer background and are interested in learning about SSL Security or how HTTPS can make a big difference in a website, this eBook will be a good choice. It covers different things that will let you know what's the importance of SSL in today's generation and what things are required by their provider to make SSL secure the website.

It begins with the history of SSL to provide the basic background of its beginning while answering other questions such as what SSL is, what's its importance and how it can impact the security of the website users if it's not installed.

It will take you through the journey of making an SSL a reality. For example, how SSL works and what is involved with it, what's the role of PKI, Ciphers, and Algorithms. It also provides information on its latest developments such as the release of TLS 1.3, the newest version and how it differs from the other previous versions.

Likewise, it will also guide you through how SSL is structured and what are the different types of SSL certificates offered by Certificate Authorities. Also, it will guide you through the management of SSL certificates and help you find common mistakes and errors made by many. Last but not least, it provides instructions on how to view SSL of a website in different web browsers and also teaches you some commonly used free OpenSSL commands that can be beneficial to anyone installing an SSL/TLS Certificate on their website.

In this eBook, I have made an attempt to distill everything about SSL certificates. It aims at encouraging even newbies to SSL certificates to understand how an SSL certificate works and its importance to websites. This "Ultimate Guide" includes detailed information on the different types of SSL certificates and will help you choose the right certificate for your website.

TABLE OF CONTENTS

Contents

SSL/TLS CERTIFICATES: THE ULTIMATE GUIDE 1

INTRODUCTION TO SSL 2

- Why SSL? 3
- Features & Benefits of SSL Certificate Security 4
- What could go Wrong, if an SSL/TLS Certificate is not Installed? 5

HOW DOES SSL WORK? 7

- SSL Architecture and Protocols 8
- Authentication 13
- Encryption 15
- Ciphers and Algorithms 19
- What Do Cipher Suites Do? 22
- Cipher Suites: Selection & Compatibility 23
- How to Know Which Cipher Suites are Used by Web Servers? 24
- Public Key Infrastructure (PKI) 24
- Key Components of PKI (Public Key Infrastructure) 25
- TLS 1.3: Faster and More Secure 29
- TLS 1.3 vs. TLS 1.2 33
- Trust Hierarchy 40

TYPES OF SSL CERTIFICATES 45

- What are the Different Validation Levels of SSL Certificates?? 45
- SSL/TLS: Structure & Formats 50

SSL Certificate Lifecycle – How To Manage SSL Certificates 54

- What is an SSL Certificate Lifecycle Management? 54
- CERTIFICATE SIGNING REQUEST GENERATION 55
- SSL VALIDATION PROCESS 57
- SSL CERTIFICATE EXPIRATION 63
- HOW TO RENEW SSL CERTIFICATE 65
- SSL REVOCATION 66

MANAGING SSL/TLS AT SCALE 73

- Features of SSL Management Tools: 73

APPENDIX A: COMMON SSL/TLS MISTAKES 74
Using Self-Signed SSL Certificates 74
Choosing an Untrustworthy Certificate Authority 75
Mistake in CSR (Certificate Signing Request) 75
Not Fully Prepared for the Validation Process 76
Problems with Your Private Key 77
Not Following the Installation Guide Properly 77
Once You Encounter a Mistake, You Don't Contact Customer Support Service 77
You Didn't Check Once the Installation Is Over 78
APPENDIX B: COMMON CERTIFICATE ERRORS 79
The security certificate was not issued by a trusted certificate authority 80
APPENDIX C - HOW TO VIEW SSL/TLS CERTIFICATES 85
How to View SSL/TLS Certificates in Different Web Browsers 85
To view SSL/TLS Certificate Details in Google Chrome (Ver. 60+) 85
To view SSL/TLS Certificate Details in Mozilla Firefox 87
To view SSL/TLS Certificate Details in Internet Explorer 90
To view SSL/TLS Certificate Details in Microsoft Edge 91
To view SSL/TLS Certificate Details in Safari 92
To view SSL/TLS Certificate Details in Chrome using Android Device 93
To view SSL/TLS Certificate Details in cPanel 94
Appendix D: OpenSSL – An Open Source SSL/TLS Tool 97
About AboutSSL.org 101
COPYRIGHT NOTICE 102

Indeed, said by an American cryptographer Bruce Schneier, “Hardware is easy to protect: lock it in a room, chain it to a desk, or buy a spare. Information poses more of a problem. It can exist in more than one place; be transported halfway across the planet in seconds; and be stolen without your knowledge.”

In today’s evolving generation, staying connected to the online world and sharing information on a website has become an integral part of our everyday life. We shop and pay bills online. Online banking and we transfer legal data or documents and share sensitive information like credit card numbers online. We do it without a second thought. Due to such advancements, threats related to these technologies are also emerging rapidly that even one slight mistake could lead to a great loss that could take ages to recover.

So, should we stop all this and go back to the good old days of doing everything manually? Well, it's one of the safest ways to stay protected. Again, it's not something we look for, nor anyone would bother to recommend. However, security is one of the crucial aspects that must be taken seriously. Whether it’s an online business or a website visitor, no one should compromise over security nor make the mistake of sharing something sensitive over an unsecured website.

Generally, communications transpire between the computer's web browser and the web server that hosts the site. Typically, when these communications are unguarded, your information is out in the open, and any interested third party can have a look at it. So, how do you deal with your information security appropriately? You can stay secure with one of the leading security protocols named 'SSL.' If you have never heard of it before, you might wonder what an SSL is, how it works and much more.

This eBook is here to answer all these questions regarding SSL. Whether you are an inexperienced person or someone who is aware of SSL, you will learn everything that will help you understand why it's so important and why it should be taken seriously.

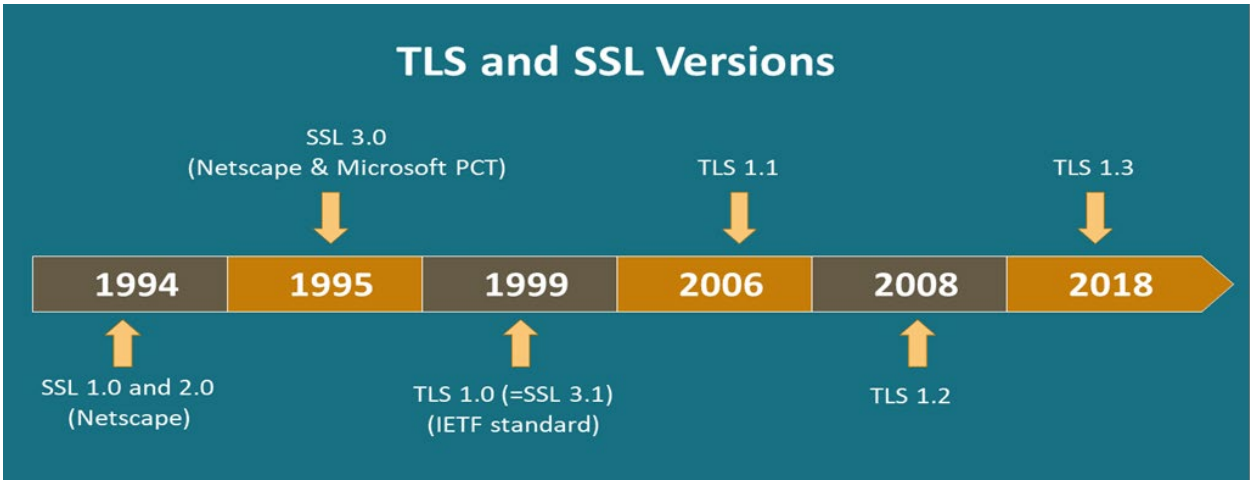
INTRODUCTION TO SSL

SSL, acronym of Secure Sockets Layer, works almost like a passport, where it serves two purposes. Firstly, it permits to access encrypted information through Public Key Infrastructure and secondly, it helps authenticate the identity of the owner of the SSL certificate.

SSL, also known as TLS (Transport Layer Security), is widely used in websites no matter what kind of website it is, whether it's a simple blog or a shopping portal. Moreover, if you have ever noticed the URLs of websites, they either show HTTPS or HTTP. The difference between the two is the 'S' at the end. HTTPS indicates that the website is secured by an SSL certificate that helps in securing the communication that happens between your web-browser and the website you're visiting.

In other words, SSL one of the security technologies that offer a secure connection between the web server and web browser. It assures that the data transferred between the web server and browser is secured and original.

Let's read its history and see how it all started.



1994 - SSL was developed by Netscape Communications when they were designing the very first version of their flagship browser.
(Many may not be aware, but the very first version of SSL was never released due to certain security issues faced during sensitive transactions such as credit card transactions, over the internet.)

Later in 1994 , Netscape made some improvements and launched the second version of SSL, SSLv2 (Deprecated in 2011), that overcame the previous problems and offered security for sensitive information. Thus, SSL became a standard protocol for the protection of HTTP based web traffic.

1995 - Netscape went further down the road to make more improvements to strengthen the cryptographic algorithms to solve the problems of SSLv2. As SSLv2 used weak MAC construction, the upgraded version called SSLv3 was released. This fixed the problem related to SSLv2 and also offered enhanced features like support for several security algorithms that were not supported previously.

1999 - In the form of upgrading SSL3.0, TLS 1.0 (Transport Layer Security) written by Christopher Allen & Tim Dierks, was defined in RFC 2246. As per the RFC, "the differences between this protocol and SSL 3.0 are not dramatic, but they are significant enough to preclude interoperability between TLS 1.0 and SSL 3.0."

2006 - TLS 1.1 was introduced in RFC 4346

2008 - TLS 1.2 was introduced in RFC 5246

2018 - Unsafe technologies in the previous versions were removed and TLS 1.3, the current version was released. TLS 1.3 offers better privacy than its predecessors.

TLS certificates are still known as SSL certificates, but the reality is that whenever someone purchases an SSL Certificate, they are actually purchasing the latest TLS certificates with options like RSA or DSA encryption. The reason is because SSL is the most commonly used term that has become more familiar among internet users.

Why SSL?

There used to be a time when SSL Certificates were more like a luxury. But in today's digital world where cyber-attacks are more prevalent, securing a website with an SSL/TLS Certificate has become more like a norm. Websites and SSL Certificates go hand in hand: if there's a website, it has to have an SSL installed! Mainly because SSL/TLS Certificates help in achieving the two main purposes of security – Encryption and Authentication.

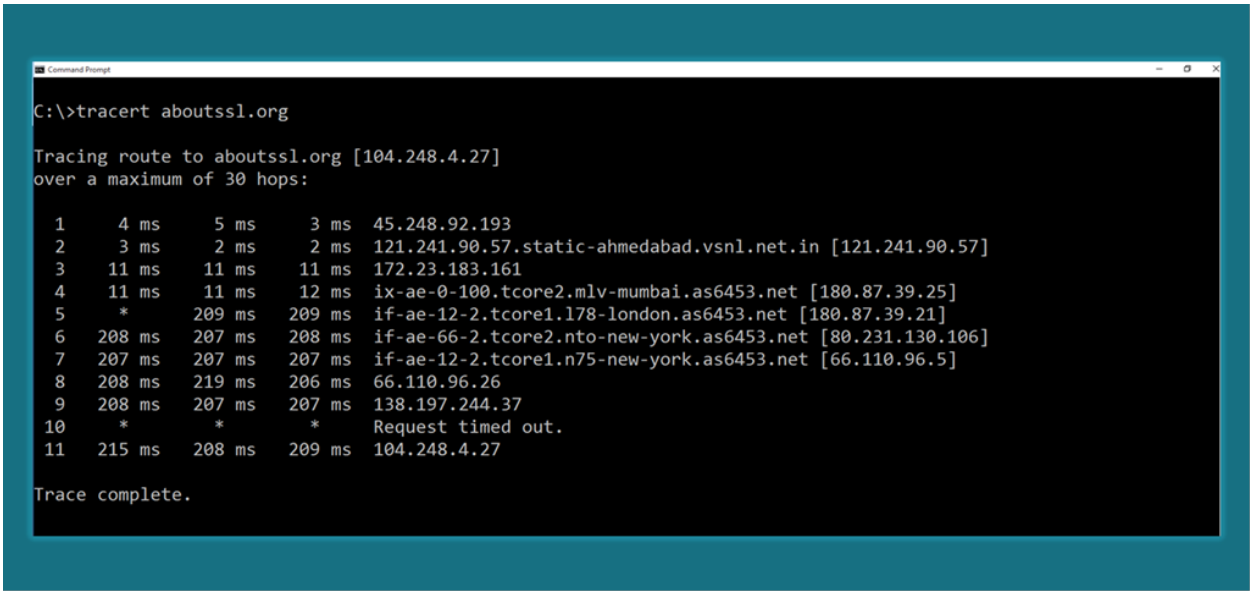
Few other reasons to have an SSL are; it helps to secure communication that happens between the website and the customer's web-browser, communication on the corporate intranet, email communications sent between the network or any private email address, transfer of information between internal as well as external servers, information sent and received between mobile devices.

Moreover, whenever someone enters any information, it goes through multiple devices before reaching its destination, which can be checked using a simple command prompt: command "tracert."

For Windows Users:

1. Go to **Start** and Click on **Run**
2. Type "**cmd**" and hit Enter
3. Once the Command Prompt opens, type a website URL. For eg. **tracert domain-name.com**
4. Hit **Enter**

You will now see a list of devices. Here, every "node" you see is a device where your sent data gets recorded before reaching its destination. Generally, 20 to 30 listings happen. This means your information has been recorded on all those devices.



Furthermore, this is exactly what happens whenever you type your sensitive information such as your credit card number, passwords, financial or any other sensitive details. Here's where an SSL/TLS Certificate comes into play. An SSL certificate encrypts such data using its complex key exchange system, hence providing a secure transaction between the web browser and the server.

Features & Benefits of SSL Certificate Security

SSL Protects Data Through Encryption

SSL/TLS Certificate protects website visitors' sensitive information which includes their login details, passwords, account details, and credit card numbers by establishing a secured encrypted tunnel between the Client (Web Browser) and the Server. SSL/TLS Certificates come equipped with strong *256-bit encryption* standard that cannot be cracked easily.

SSL Confirms Your Identity

It's not easy to trust someone on the internet, nor can you be sure that the website you are visiting is genuine. Many fake websites dupe their visitors and gain sensitive information by using someone else's name. SSL certificates can help avoid such situations. The first step in getting an SSL Certificate is to complete the verification process. SSL providers verify the legitimacy of the person or the organization purchasing the SSL certificate through a process that is done based on certain rules and regulations set by CA/B, a joint forum of CAs and Web browsers.

SSL Offers Non-Repudiation

Combination of Encryption, Integrity and Authentication establishes non-repudiation. This means none of the parties in a secured transaction can legally say that their communication has come from someone else. SSL removes the option of a party to repudiate or in other words "take back" information which has been communicated by them online.

Helps in Avoiding "Not Secure" Warning Message

Earlier in 2014, the leader of Search Engines, Google, declared that websites that have an SSL/TLS Certificate installed would get a little more preference in search results, as a part of their mission to make the entire web safe and encrypted. Since 2018, Google Chrome and other popular web browsers like Mozilla Firefox have taken one more serious step towards it. With the release of their latest web browser versions, they even started displaying the "Not-secure" warning on non-SSL websites. Some websites even fail to load on these popular browsers. So, if you want your visitors to visit your website, it's mandatory to have an SSL.

What could go Wrong, if an SSL/TLS Certificate is not Installed?

On the contrary, if an SSL/TLS Certificate is not installed, attacks like APT (Advanced Persistent Threats), Man-in-the-middle (MIM), and Protocol attacks can happen easily.

APT (Advanced Persistent Threats)

As the name implies, it is an "Advanced" attack, which means attacks carried out by those who are proficient in technology and well-funded by an external entity. Various techniques like drive-by-downloads and Microsoft SQL injection are also used.

Moreover, it is a targeted attack, done with a purpose. However, it does not mean all targeted attacks are APT, as it uses customized attacking methods such as zero-day vulnerability exploits, viruses, worms and many other techniques. It is a type of attack that takes time and often it is noticed after a very long time, i.e., after the damage is done.

Man-in-the-middle (MITM) Attacks

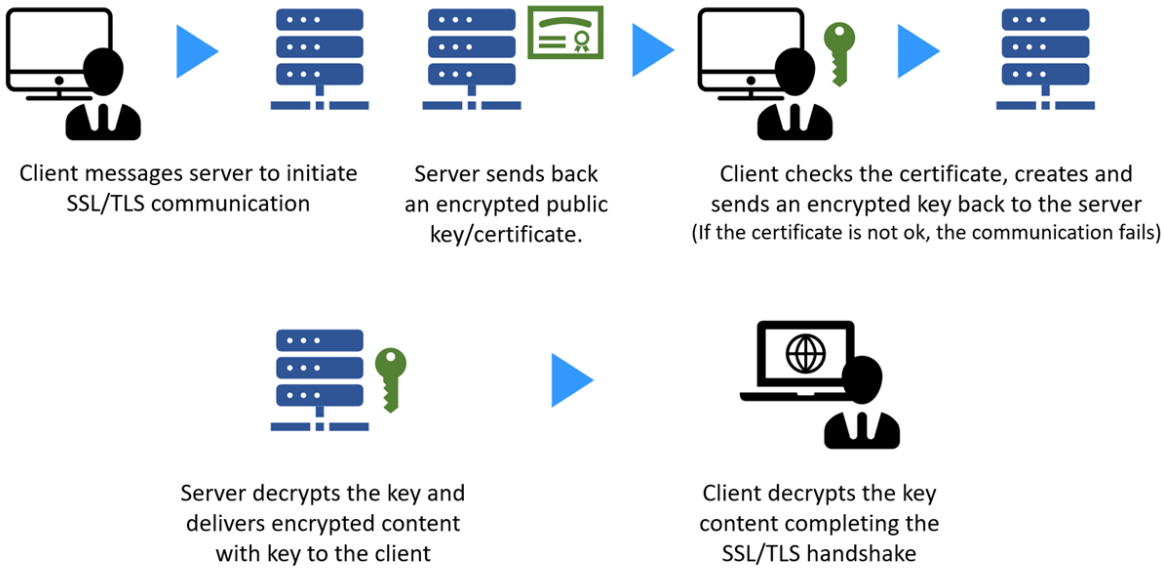
MITM is a serious problem, which is a threat to authentication capabilities. It's one of the attacks where an attacker is capable of secretly changing the data between two people who think they are directly communicating with each other and it is quite challenging as it is done remotely with a fake address. In other words, it is one of the attacks that takes place when two systems are intercepted by an unauthorized entity.

Protocol Attacks

It is one of the attacks that focuses on the resources of the server. Attacks like Slowloris and HTTP Flood are examples of this kind of attack. HTTP flood is an attack which takes place by portraying fake GET or POST requests, making it appear as legit. Additionally, less bandwidth is used for HTTP flood attack, which forces the server to use maximum resources. On the other hand, Slowloris an invention of *Robert "RSnake" Hansen*, is a denial-of-service attack tool which is used to attack a web server with minimal resources. In other words, Slowloris tries to keep multiple connections open with HTTP flooding. The desired target is connected with the partial request of HTTP header that never completes resulting in denying of any additional connection attempt from clients, due to attack it's filled with the maximum concurrent connection pool.

HOW DOES SSL WORK?

As we learn how SSL/TLS certificates are one of the essential parts of the data encryption process to make internet transactions safe and secure, let's now learn how the SSL/TLS Certificate works.



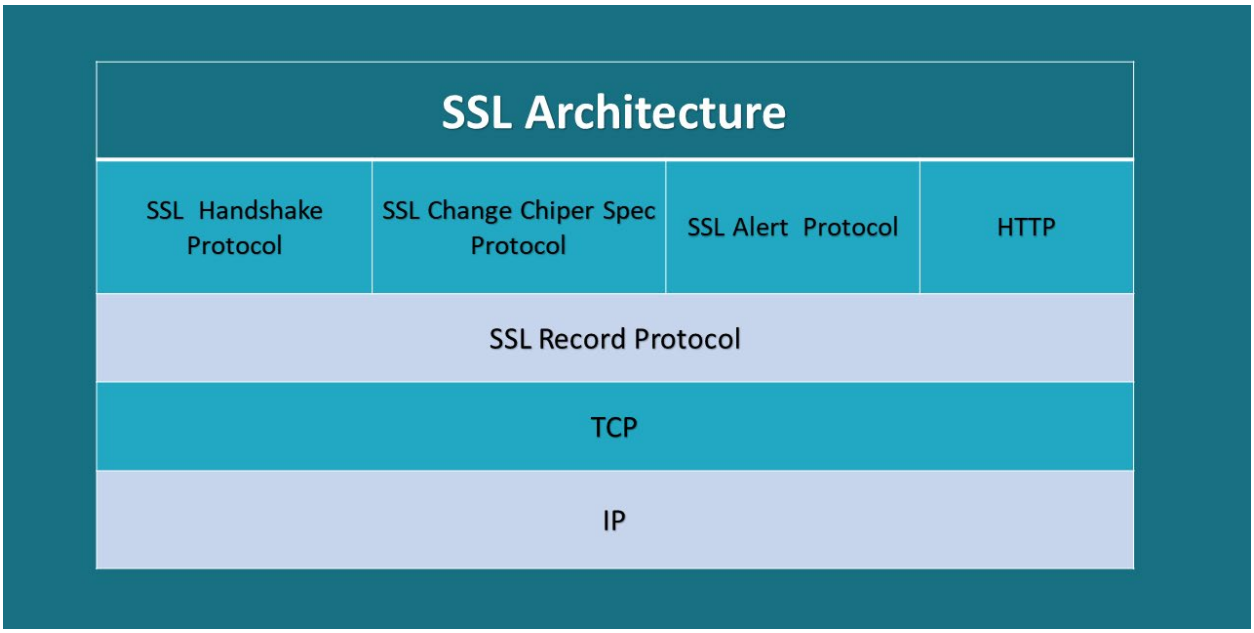
As per a layman's standpoint,

- ✓ First, the server or browser tries to connect with an SSL secured website (i.e., web server). The server or browser makes a request to identify the web server.
- ✓ The web server sends a copy of the SSL/TLS Certificate to the web browser or server.
- ✓ The web server or browser checks whether the SSL certificate is trustworthy or not. If it is trustworthy, it sends a message to the web server.
- ✓ To start an SSL encrypted session, the web server sends back a digitally signed acknowledgment.
- ✓ Finally, the encrypted session begins and the encrypted data is shared between the server/browser.

Also, the working of an SSL may look like a seamless process. But there are several things working in the background that make the SSL connection successful. For example, what are the different types of encryption, how the authentication of the message is done, what ciphers & algorithms are used and much more.

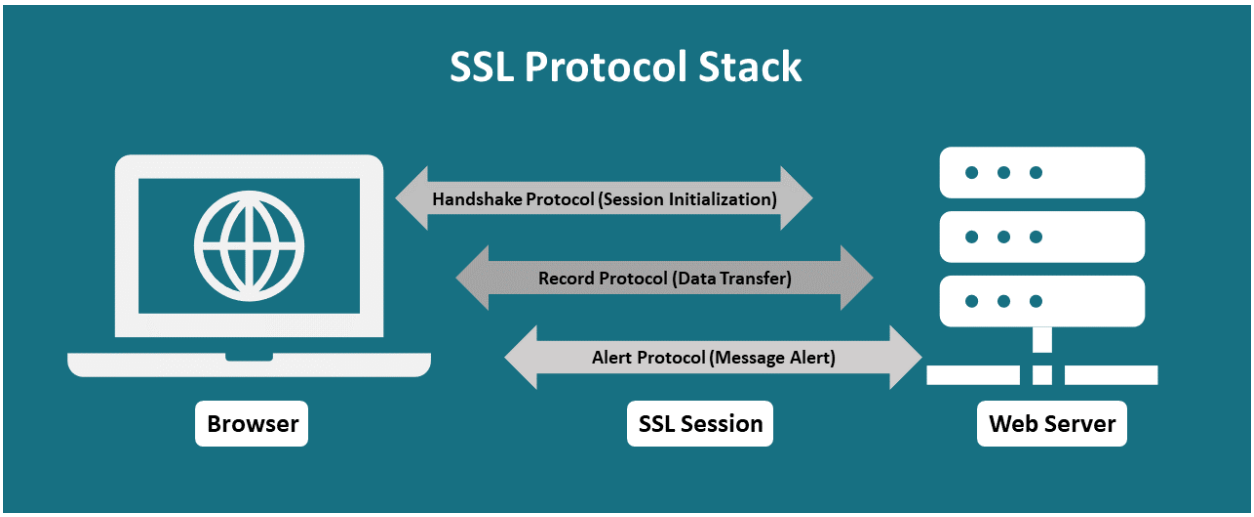
SSL Architecture and Protocols

SSL/TLS is a protocol which operates directly on top of the TCP, (though there are some other implementations for a datagram-based protocol like UDP.) This is the reason how protocols of higher levels are left unchanged while providing a secured connection (for example, HTTP). One thing to note is that HTTP is identical to HTTPS beneath the SSL layer.



If SSL/TLS is installed correctly, attackers will only be able to see which Port and IP you relate to, how much data is sent, which encryption and compression are used. Moreover, an attacker can terminate the connection, but the positive thing is that both the sides will get to know that the connection has been interrupted by a third-party.

SSL involves two entities, Server and Client. Here, Client is the one that initiates the transaction and the other entity Server is the one that responds to it. (Client is the Web-browser and the Website server is the Server.)



SSL is designed to make use of TCP to offer a reliable end-to-end secure service. And, it's not a single protocol but a two-layered protocol, where the SSL Record Protocol offers basic security services to different high-level protocols. Specifically,HTTP, which offers a transfer service for the interaction between the Web client and the server, can operate on top of SSL. Other three higher-layer protocols are defined as a part of SSL, namely, the Alert Protocol, the Handshake Protocol and the Change CipherSpec Protocol, that are used to manage SSL exchanges. Let's understand each of them.

SSL Handshake Protocol

SSL Handshake Protocol is the technical name given to the process which establishes an HTTPS connection. Almost all the important work is done here in this protocol. In other words, it establishes a secure channel between the server and the client.

The main purpose of the SSL handshake protocol is to perform all the needed cryptographic work for having a secure connection such as checking the authenticity of the SSL certificate, are they created and signed by a trusted Certificate Authority, proving that the private key owned by a server is associated with the certificate and this entire SSL handshake is done within few hundred milliseconds. Moreover, SSL handshake is the first thing that happens in an HTTPS connection, even before the webpage loads completely.

Every software is different from one another. Due to this, the first step in the handshake protocol allows the client and server to share their capabilities to find out the mutually supported cryptographic features. For example, Web browsers are one of the typical clients, but their features differ depending upon the browsers like Microsoft Internet Explorer, Google Chrome, and Mozilla Firefox. Similarly, when it comes to the server, like Windows Server, Apache and NGINX, their features are different from each other.

Though one thing to note that SSL Handshake is a series of several steps that is done to achieve the following three main tasks.

- ✓ Exchange of Encryption Capabilities
- ✓ Authentication of the SSL/TLS Certificate
- ✓ Exchanging or Generating a Session Key

If you are interested in understanding what the exact process is, below are the complete illustrated steps. (Note that in TLS 1.3, the forthcoming protocol version, the handshake design has changed. Thus, these steps are related to TLS 1.2.)

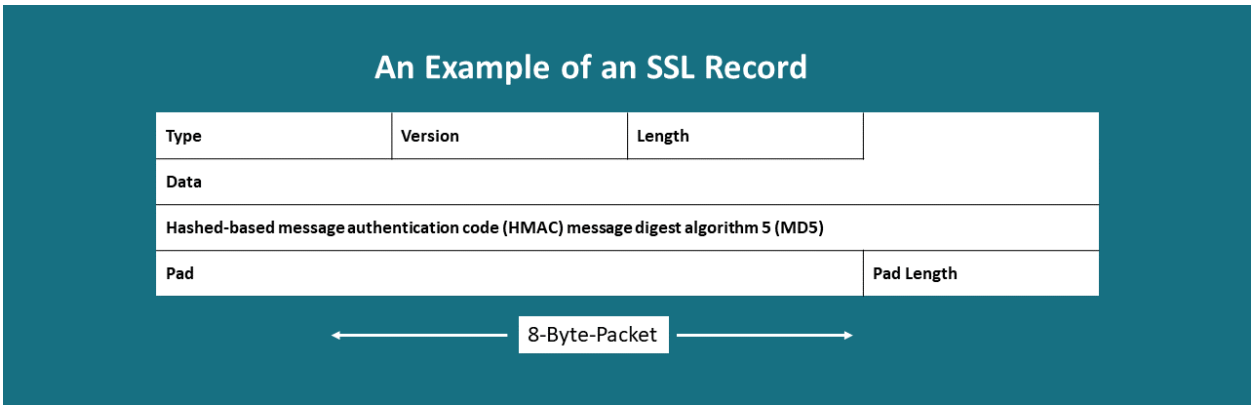
Step	Client	Direction	Message	Direction	Server
1			Client Hello	>	
2		<	Server Hello		
3		<	Certificate		
4		<	Server Key Exchange		
5		<	Server Hello Done		
6			Client Key Exchange	>	
7			Change Cipher Spec	>	
8			Finished	>	
9		<	Change Cipher Spec		
10		<	Finished		

Step No.	Message	Action
1	ClientHello	This is the very first step. Here the client initiates the handshake by sending a message "ClientHello," which recommends the parameters of SSL, that will be used during the entire SSL session.
2	ServerHello	Here, the server responds to the client with the message "ServerHello," containing the selected SSL parameters from the provided list that will be used during the SSL session. If the client and server fail to share common parameters, the connection will be terminated right there.
3	Certificate	Here, the Server will send the SSL Certificate chain (it includes leaf and intermediate certificate) to the Client. Then, the Client will start checking whether the certificate is legitimate by verifying the digital signature of the certificate and the certificate chain and checking if there's any potential problem with the certificate data (whether the certificate is expired, wrong domain name, etc.). The client will also make sure that the server possesses the private key of the certificate and this entire process is done during the key exchange/generation.
4	ServerKeyExchange	It's an optional message, which is needed when certain key exchange methods ask the server to provide additional data.
5	ServerHelloDone	Here, a part of the SSL negotiation are concluded by the Server. In other words, this message "ServerHelloDone" tells the Client that all the messages have been sent over.
6	ClientKeyExchange	Here, the client sends the information regarding the session key, which was encrypted using the server's public key.
7	ChangeCipherSpec	Here, the Client provides instructions to the server that it activates all the negotiated SSL parameters for all future message it sends.
8	Finished	The Client instructs the server to verify whether the SSL negotiations have been successful or not.
9	ChangeCipherSpec	Here, the Server gives instructions to the client to activate all the negotiated SSL parameters for future messages it sends.
10	Finished	Finally, the Server instructs the client to verify whether the SSL negotiations are successful or not.

Completion of the above-mentioned steps indicate completion of the SSL handshake. Both parties will now have a session key and will begin to communicate with an encrypted and authenticated connection. At this point, the first bytes of "application" data (the data belonging to the actual service about which the two parties will communicate – i.e., the website's HTML, Javascript, etc.) can be sent.

SSL Record Protocol

The SSL Record Protocol is responsible for handling encryption of all messages. This protocol offers a common format used to frame all the messages of the Handshake, Alerts, ChangeCiperSpec and Application Protocols. In other words, it offers basic security to other higher layer protocols, namely Handshake Protocol, Change Cipher Spec Protocol and Alert Protocol.



This protocol consists of the summarized data, message type, version, length and digital signature, which makes it 8 bytes long. However, sometimes there's a possibility of a frame having a padding and padding length as there is a fixed record length.

SSL Alert Protocol

This protocol provides SSL related alerts, as it is responsible to handle questionable type of packets.

Generally, it handles three different types of alert messages:

- ✓ Warning
- ✓ Critical
- ✓ Fatal

Here, the session is further restricted depending on the received message (i.e., warning or critical) or else terminated (fatal).

The ChangeCipher Spec Protocol

The change cipher spec protocol occurs for signaling the transitions in cipher strategies. It contains a single message which carries a single value byte of 1. The purpose of this protocol is to compress and encrypt that message under the connection state. Likewise, the message of this protocol is sent by both the party

server and the client for notifying the receiver that successive records will stay protected under the newly exchanged CipherSpec and keys.

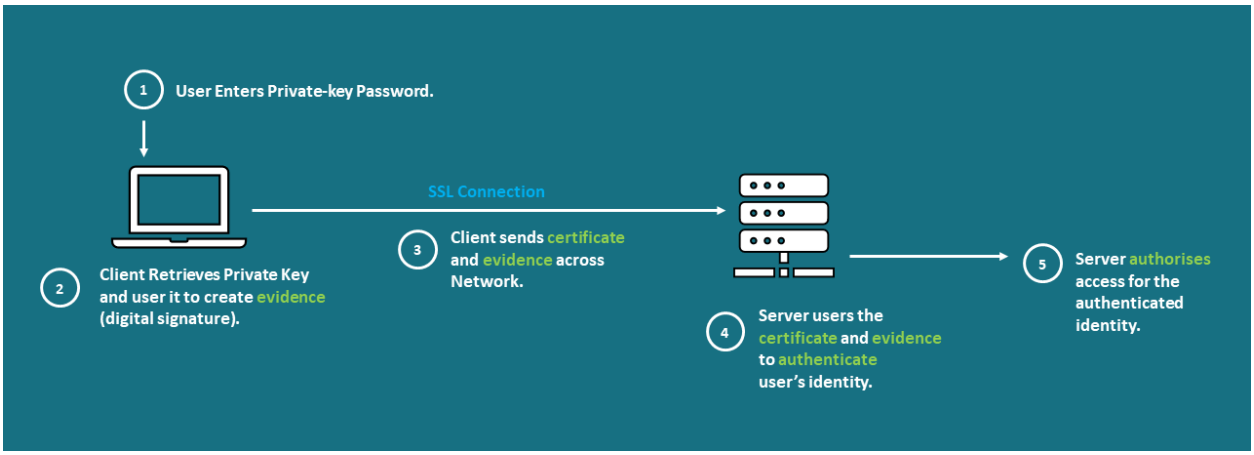
Authentication

Authentication is one of the simplest terms of identity confirmation. It is one of the processes used for network interactions and it also involves the recognition of one party by the other. Moreover, there is more than one way to use authentication over networks and certificates are one of those.

Network communications are generally done between two clients — for example, a web browser and a server. Here, client authentication is the identification of a client by a server and server authentication is the identification of a server by a client. (For example, the client is the person who is using a software/web browser and the server is the organization running their server at the network address.)

Server and Client authentication are not limited to the form of authentication supported by the certificates, but it also ensures nonrepudiation. For example, an email message with the digital signature combined with the certificate identifies and authenticates the sender of the message and at the same time makes it difficult for the signer to tell that the email is not sent by that person.

One thing to note is that Client authentication based on certificates are part of the SSL protocol. Here, the client digitally signs a piece of data generated randomly and sends that signed data as well as the certificate to the network. Lastly, the server confirms the validity of the certificate after validating the signature.



Below are the steps that explain how authentication of a Client by a Server is done:

1. The client software manages a database which consists of private keys corresponding to the public keys published in all the certificates issued for that client. Here, the client asks for the password of the database to access it for the first time for any given session. For example, the user attempting to access an SSL-enabled server requires authentication for the certificate-based client.

   Note: Once the password is entered, it will not be asked again during the whole session, even to access other SSL-enabled servers.

2. After that, the Client unlocks the database consisting of the private-key and retrieves it for the user's certificate and uses that private key for signing data which is generated randomly from the input made by both the server and the client. In return, the data and the digital signature work as evidence that the private key is valid. Additionally, a digital signature is only created with a private key and it's validated with the corresponding public key against the data which is signed and it stays unique to the SSL session.

3. Randomly generated data as well as the user's certificate, are both sent across the network by the client

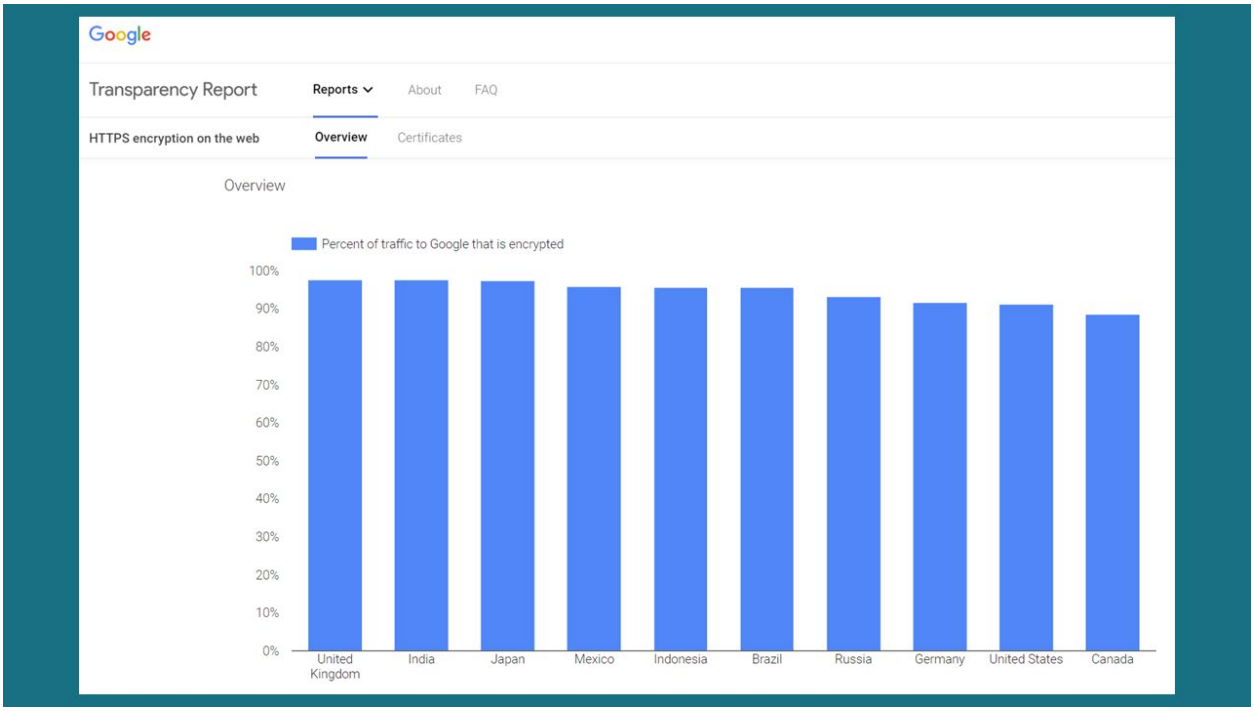
4. To authenticate the identity of the user, the server uses both the signed data as well as the certificate.

5. The server may also perform other authentication tasks like checking whether the certificate is stored in an LDAP directory in the user's entry, presented by the client. And once it is checked, it also evaluates whether the identified user is allowed to access the requested resource. Furthermore, the evaluation process may use other authorization mechanisms, maybe company databases or information consisting of an LDAP directory. If the result of the evaluation is positive, then the server will allow the client to access the requested resources.

Here, the authentication part of the client and the server interaction is replaced by the certificates. Rather than requesting the user to send passwords through the network frequently, the single sign-on feature is used. Here, the user enters the password of the private-key database for the first time, without sending it through the network. Once it's done for the whole session, the user's certificate is provided by the client to authenticate every new server encountered by the user. Lastly, existing authorization is not affected, which was founded on the authenticated user identity.

Encryption

Encryption is one of the essential applications of cryptography. It is used to make data incomprehensible to ensure confidentiality. Also, encryption has become an integral part of many products and services. It's also widely used over the internet by using modern encryption technologies such as SSL/TLS.



Percent of Encrypted Request Google Receives Worldwide

Encryption Behind SSL/TLS Cryptography

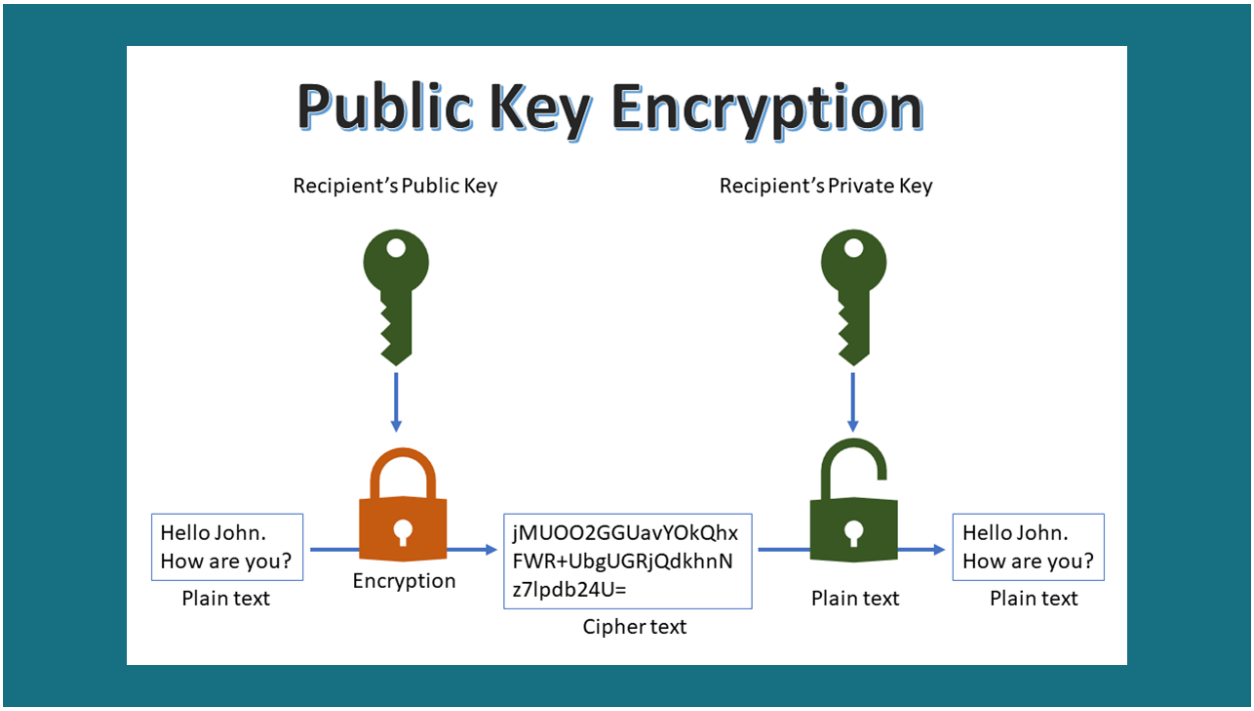
SSL/TLS is one of the standard security technologies that offer an encrypted link between a client (browser) and a server (website) or a mail client like Outlook, to transmit sensitive information like Credit Card details, Login Details, Social Security Numbers safely.

But how it's all achieved? Let's understand the technology that works behind SSL encryption to provide a secure connection, namely Public Key (Asymmetric) encryption and Symmetric Key encryption.

What is Public-Key Encryption and How does it Work in Certificates?

Public-key encryption, also called as Asymmetric Encryption is one of the encryption schemes that uses two different keys namely a public key and a private key. The Public Key will be accessible to everyone as it's public and Private Key will remain private with the owner of that key. Though these keys are mathematically related, they are not identical. Here, both the keys

are used for different purposes. The public key is used to encrypt the data, whereas the Private key is used to decrypt it.
Likewise, Asymmetric encryption uses certain encryption algorithms such as RSA & DSA to create the public and private keys that are based on the difficulty of mathematical problems. However, from a computational point of view, it's quite easy to make the public and private keys and to encrypt with the public key and decrypt with the private key. But, it's almost impossible for anyone to derive the private key based on the public key.



Furthermore, Public Key Encryption helps in authenticating and exchanging keys. For example, whenever a user (client) visits a website (server), it will send a "clientHello" message, which consists of a list of cipher suites supported by it in an ordered preference. And all of these are encrypted with the public key of the server. Now, the server makes use of its private key to decrypt the message clientHello and responds with the serverHello message with its certificate, a chosen cipher suite and it's key. Once the serverHello message is received by the client, the client and the server begin their communication with the symmetric encryption key exchanged by them.

Note: Session keys often change and many times, a different session key is used for each message.

Bulk/Symmetric Encryption

As we saw earlier, to confirm the genuineness of the server, the authentication process takes place through the Public Key Encryption (Asymmetric Encryption) and Digital Signature. But, once the server authentication is done, for the rest of the session, the Client and the Server make use of Bulk/Symmetric- Key encryption, for encrypting all the exchanged information and detection of any tampering.

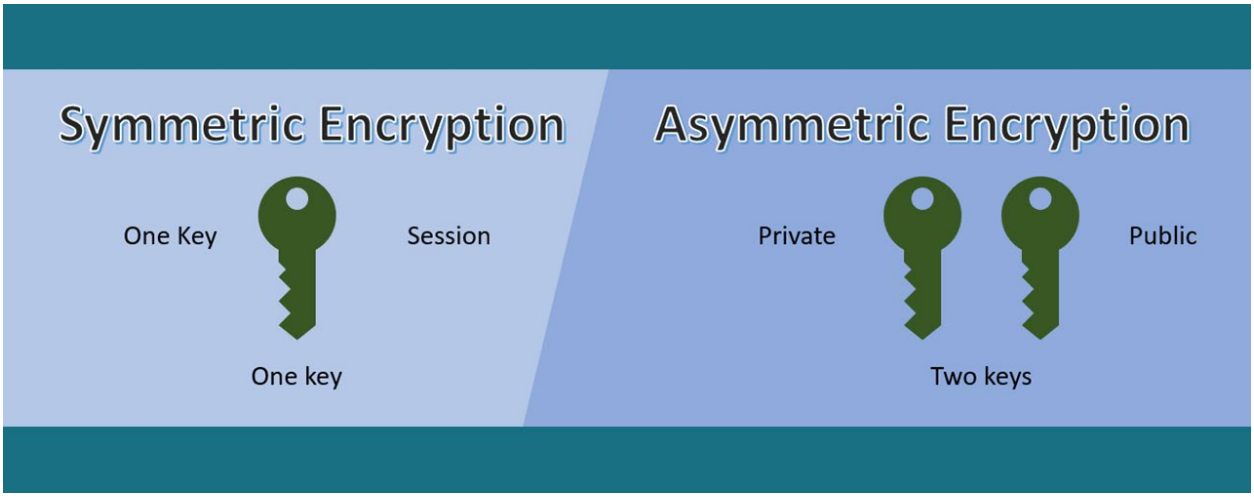
In other words, asymmetric encryption is used at the time of the SSL Handshake as a verification method, where the browser and the server negotiate an encrypted connection and exchange Session Keys.The session keys use symmetric encryption to further communicate during the entire secure session.

Let's understand the difference between the two:

- ✓ Asymmetric Encryption
- ✓ Bulk/Symmetric Encryption

What Is Bulk/Symmetric Encryption and How It Differs from Asymmetric Encryption?

Symmetric encryption is the encryption method which involves one secret key to cipher and decipher the information. It's one of the old techniques, which uses numbers, words, or strings of random alphabets as a secret key.
However, to encrypt and decrypt the message, the recipient must be aware of the secret key. Some examples of symmetric encryption algorithm are Blowfish, AES, DES, RC4, RC5 and RC6.Among them, the most widely used algorithms are AES-128, AES-192, and AES-256.



On the other hand, Asymmetric encryption, also known as Public Key Cryptography is a new method compared to Symmetric encryption. It uses two different keys for encrypting plain text.

Here, secret keys are exchanged over the large network or the Internet while ensuring it will not be misused. The public key is available to anyone who wants to send you a message and the second private key is kept secret only with you.

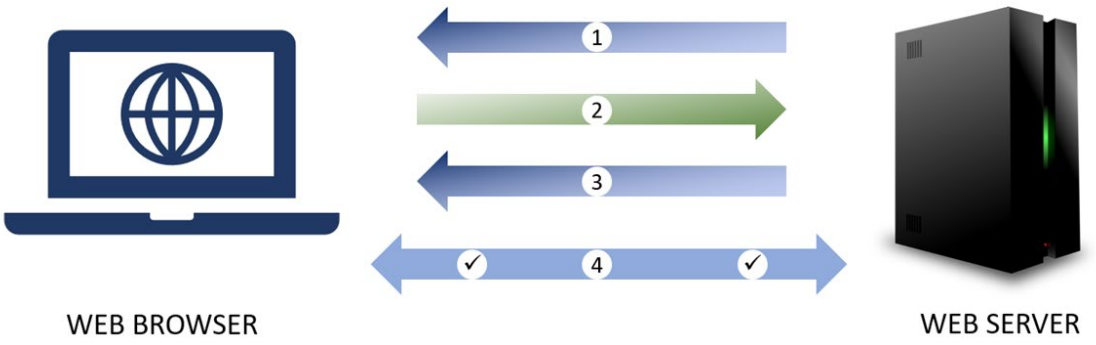
Besides, a message encrypted via public key can only be decrypted through its private key and a message encrypted through a private key can only be decrypted with its public key. However, the public key does not require any security and it's available to anyone over the internet. Lastly, Asymmetric encryption is one of the most widely used encryption for daily communication over the internet. Some of the well-known asymmetric key encryption algorithms are ElGamal, RSA and DSA.

Quick Look at the Differences Between Asymmetric & Symmetric Encryption

- ✓ A single key is used by symmetric encryption and it's shared among people who are looking to receive the message whereas asymmetrical encryption uses a pair of keys called public and private keys for encrypting as well as decrypting the message whenever the communication takes place.
- ✓ Comparatively, Symmetric encryption is an old method, whereas Asymmetric is new.
- ✓ Asymmetric encryption was invented to overcome shortcomings of the Symmetric encryption model of sharing the key. Asymmetric encryption eliminates the sharing of the key by making using of a pair of public-private keys.
- ✓ Asymmetric encryption is more time consuming compared to Symmetric encryption.

Asymmetric & Symmetric Encryption: How SSL/TLS Uses Both?

PKI (Public Key Infrastructure), the set of policies, procedures, hardware, software and people are important for creating, managing, using, storing, distributing and even revoking digital certificates. Using Certificate Authority (CA), PKI binds keys with user identities. For example, SSL Certificates contain an asymmetric public key and a private key pair.



The session key created by the server and the web browser during the SSL/TLS Handshake is symmetric. Let's understand it in detail.

- ✓ Copy of asymmetric public key is sent to the Server.
- ✓ Once the copy is received, the symmetric session key is created by the Browser while encrypting it with the asymmetric public key and sends it back to the server.
- ✓ To get the symmetric session key, the Server uses an asymmetric private key to decrypt the encrypted session key.
- ✓ Now, Server & Browser, along with the symmetric session key encrypt and decrypt all the transmitted data. Lastly, it allows a secure channel as only the server and browser are aware of that symmetric session key and it's used only for that session. If the browser wants to repeat the same session with the server the next day, a new session key will be created.

Ciphers and Algorithms

If you have heard or interacted with HTTPS encryption or SSL/TLS, then you might have come across "cipher suites" too. Though it sounds like some fancy name, it's a fact that it's one of the critical things that many people are not aware of it. So, what are ciphers? What are the cipher suites?

First, let's see what ciphers are and then we will get into details of cipher suites. Put simply, ciphers are algorithms. To be more precise they are a set of instructions to perform a cryptographic function which can be encryption, decryption, digital signatures or hashing.

As the years pass by, ciphers have changed a lot and become more and more complex. But the concept behind ciphers is still the same. Whether it's the first historical Cipher of Caesar, the notorious Enigma cipher of World War II or any algorithm of today's date, the idea is still the same; encoding or enciphering a message in a way that only the intended party or a person can read it.

Cipher Suite: What is it and What Makes it so Important?

A cipher suite is a set of algorithms used to help determine the security settings at the time of the SSL/TLS handshake. When the ClientHello and ServerHello messages are exchanged, firstly the client has to send a prioritized list of supported cipher suites and then the server responds with its selected cipher suite.

In other words, cipher Suite is a set of information which helps in letting know how the web server will communicate data securely over HTTPS, SMTP, FTPS and other network protocols.

Here, the web server uses some of the algorithms and protocols to know how it will secure the web traffic.

A typical cipher suite looks like below.

ECDHE-ECDSA-AES128-GCM-SHA256

Some of the things that depend on cipher suites are:

- ✓ The security level of the HTTPS traffic, where your safety and that of your website visitors are considered.
- ✓ The compatibility of HTTPS traffic to decide who can see warning messages, errors or other issues related to it.
- ✓ The performance of your HTTPS traffic, such as how fast your website loads on the user's device or what is the page speed of your website.

Cipher suites are a blend of different things just like recipes that are made with different ingredients. For example, to make banana bread, we need certain ingredients like:

FLOUR-BANANAS-EGGS-SUGAR-BUTTER

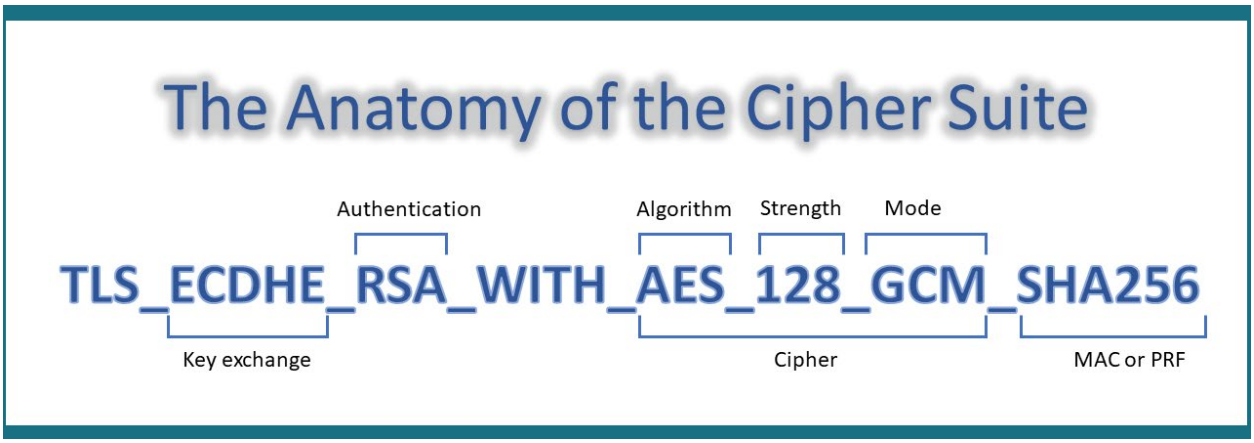
Same as the above, to provide a successful and secured connection, cipher suite needs different ingredients that are protocols and algorithms instead of food items.

- ✓ ECDHE is a type of Key Exchange Algorithm
- ✓ ECDSA is a type of Authentication Algorithm
- ✓ AES128 is a type of Bulk Encryption Algorithm
- ✓ SHA256 is a type of MAC Algorithm

If your web server uses HTTPS, then it will have a list of cipher suites separated by a colon (:) that looks like:

ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA256

So, after binding it with TLS, its anatomy will look like the below the example where TLS is similar to the latest version, TLS 1.3:



What Are Cipher Suites Made Of?

Below are the four different components based on which cipher suites are made:

1. **Key Exchange Algorithm:**
   To assure confidentiality at the time of data transmission through different secure file transfer protocols such as HTTPS, the data needs to be encrypted. To get this process in action, two communicating parties need to have a shared key that can encrypt and decrypt the data. This is achieved with symmetric encryption. But Symmetric encryption comes with a weakness. For example, if the shared key becomes accessible to attackers, they can easily decrypt all the data. To avoid such situations, the industry has developed certain key exchange protocols. They are the key exchange algorithms such as ECDHE, ECDH, DHE, and RSA that help in securing the exchange of symmetric keys over the networks.

2. **Authentication Algorithm:**
   To ensure secure and correct data transmission, a web server has to verify the identity of the user who will have access to the data. Basically, in this process, a user is asked to provide credentials that include a username and password. To keep this authentication process secure, cipher suites have an authentication algorithm such as ECDSA, RSA, and DSA.

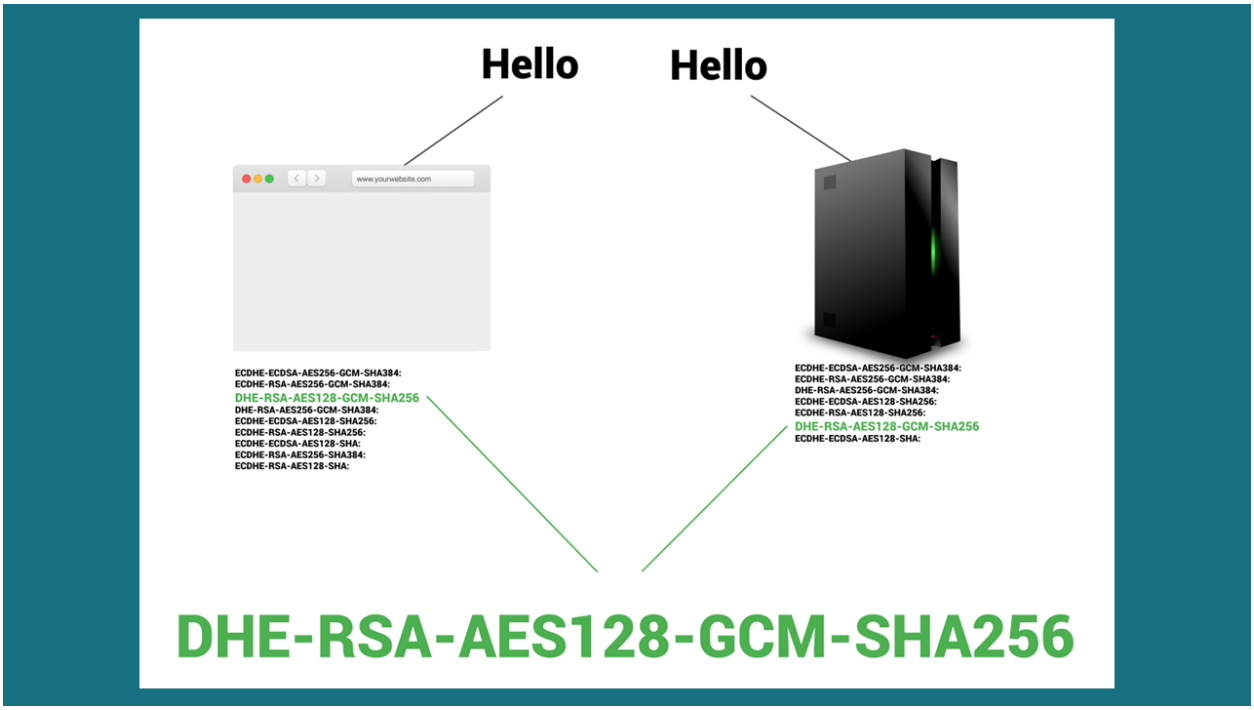
3. **Bulk Encryption Algorithm:**
   To ensure the secure transmission of data, cipher suites offer a bulk data encryption algorithm. Some of the widely used algorithms that come under this category are CAMELLA, AES, and 3DES. Moreover, as per Microsoft, the key of bulk encryption is generated by hashing one of the MAC keys with CryptHashSessionKey along with the content of the message and other data.

4. **MAC (Message Authentication Code) Algorithm:**
   MAC (Message Authentication Code) algorithm is one of the algorithms, which is used for the verification and authentication of the received message. Here, both the sender and the receiver share a common key to make the MAC algorithm work. But it does have a drawback, as it's not capable enough to protect the message even if any intentional change occurs in the authentication. Due to this, it is possibile for an intruder to make a change in the message while calculating the new checksum and replacing it with the original one. To avoid such situations, CRS (Cyclic Redundancy Check) algorithms are used which can help at a certain level by randomly detecting damaged parts of the messages. However, it cannot detect conscious damages done by an attacker. Some of the common algorithms are MD5 and SHA.

Note: GCM (Galois Counter Mode), is a type of cipher mode, a block of operation which uses universal hashing over a binary Galois field to offer authenticated encryption and authenticated decryption. In other words, it's a mode of operation for symmetric key cryptographic block ciphers widely used due to their performance and efficiency — for example, running AES (Advanced Encryption Standard) in GCM with 256-bit keys.

What Do Cipher Suites Do?

As we saw earlier, cipher suites are one of the integral parts of how the website functions over security protocols like HTTPS with the algorithms, which help in the process of data security. But in today's date many computers with different operating systems and versions are available, where different web browsers are also used. So, there has to be a way to adjust all these combinations. Here, cipher suites help. While making a secure connection, a web browser and a web server can compare their lists of cipher suites and use the compatible one. Also, it's an essential part of the "handshake" which happens during the connection between server and browser.



At the time of Handshake, the server and the client exchange a list of prioritized cipher suites and decide to use the one which is best supported by both.

The cipher suites that are on the web server help in determining how secure, fast and compatible your HTTPS traffic will be. Moreover, it's important to know that a cipher suite used by a web server affects several things like security, speed, and compatibility that are essential and helpful for webmasters to make improvements by adjusting used cipher suites.

Cipher Suites: Selection & Compatibility

The web server has many cipher suites, most of the time a collection of more than a dozen cipher suites to ensure compatibility. But not all web servers and web browsers will support all the listed cipher suites. As a solution, both the browser and the server compare their cipher suite list and based on that, they decide which one will be used before moving further. And, sets of cipher suites assure that the most secure and compatible cipher suite will be used to provide the best security.

Also, Mozilla, the inventor of the Firefox web browser, provides a resource for the recommended security configurations, including sets of cipher suites further divided into the following three different categories.

- ✓ **Modern Compatibility**
  It's for the clients that support TLS 1.3 and do not need backward compatibility. It provides an extremely high level of security. The cipher suites listed in this category are

the latest ones, which can be used by anyone who do not expect any website visitors who rely on any older machines or web browsers.
For example, a tech blogger, may not run into a problem for using the modern cipher suites as mostly their users will already be tech-savvy who will most likely stay updated.

- ✓ **Intermediate Compatibility**
  It has a default set of cipher suites compatible with most websites.
  For example - Services that do not want compatibility with legacy clients like WinXP, but still want to support various types of clients compatible with Chrome 1, Firefox 1, and Safari 1.

- ✓ **Old Backward Compatibility**
  It's a set of cipher suites that are not advisable to use but could be used as a last resort for websites which rely on users who have older machines, operating systems and older software.
  For example - With Clients using Windows XP/IE6

How to Know Which Cipher Suites are Used by Web Servers?

If you are not aware and want to know which cipher suites are supported by your web server, then you can find it with the help of free tools like SSL Server Test where you just have to enter the hostname and click the button. Once the process is complete, you will be provided with the result in the ordered preference the cipher suite is used by the server. Below is the screenshot of the same:

Public Key Infrastructure (PKI)

In today's date, if we can safely communicate with people whose public keys we have is because of Public Key Cryptography. But still, there are many problems that need to be addressed, such as how to store a public key or revoke them. Most importantly, how to do it globally when we have millions of servers, people and devices? The answer is the Public-Key Infrastructure (PKI).

What is Public Key Infrastructure?

For most people, PKI means the public-key infrastructure as often seen on the Internet. However, the meaning of PKI is much broader because its development is originally for other uses. Also, the term “PKI or Internet PKI” was introduced by PKIX, a working group of IETF (Internet Engineering Task Force) for using it on the Internet through X.509, where it focuses on how browsers validate and consume certificates.

In other words, PKI (Public Key Infrastructure) is a set of rules, procedures and policies that are needed to create, distribute, manage, store, use or revoke certificates and manage public-key encryption.

Apart from this, PKI is based on asymmetric encryption and it is majorly used to secure electronic communication for email, internet banking, online shopping and also communications of millions of users and the website to which they connect with the help of HTTPS.

Key Components of PKI (Public Key Infrastructure)

A typical PKI environment includes the below components.

Certificate Policy
Fundamentally, the certificate policy is the security requirement used to define the hierarchy and structure of the PKI environment. Also, the policies bounded with the handling & management of keys, revocation, profiles & formats of the certificate, secure storage with many other details.

Root Certificate Authority (CA)
As its name implies, root CA is an entity which is the "main root of trust" in the implementation of PKI and it's accountable for identity authentication in the PKI environment.

Subordinate or Intermediate CA
The intermediate or subordinate CA is certified through a root CA to use it specifically as per the definition provided by the certificate policy. Additionally, digital certificates are signed and issued by sub-CAs.

Certificate Database
As its name implies, it's a database of certificates which stores the records of a certificate.

Revocation Services

They are the servers who post updated CRLs (Certificate Revocation Lists) or OCSP (Online Certificate Status Protocol) responders which makes use of CRLS to respond to revocation lookup checks for all the devices that cannot process CRLs by their own.

Digital Certificates

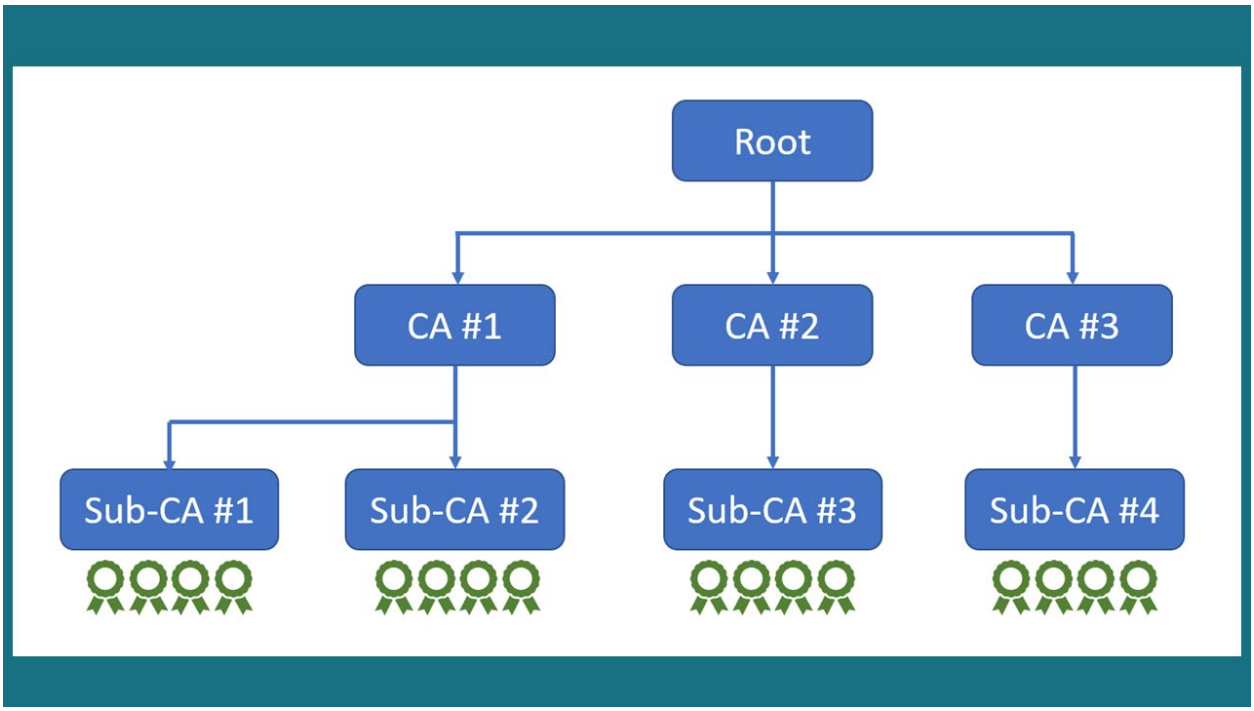
It's a type of digital identity which is embedded with a device which offers security and authentication between servers and the devices while granting access to resources. It's generally issued by the sub-CA.

PKI Hierarchy

PKI hierarchy or Hierarchical PKI is one of the PKI trust models. PKI hierarchy can have one or multiple tiers. If it's a single-tier PKI environment, the Root CA will only be your CA server. If it's multiple tiers, the Root CA will issue other subordinate CA certificates right below the root and there won't be any need to access the Root CA server daily. All the certificates can be requested by users from the subordinate CA itself while letting the Root CA offline.

Likewise, having the Root CA offline will also increase the security of the PKI environment as no one will have network access to the server. Lastly, how many tiers can be used depends on what your goal is with the PKI environment, the requirements of security or the trust you want to keep in the environment.

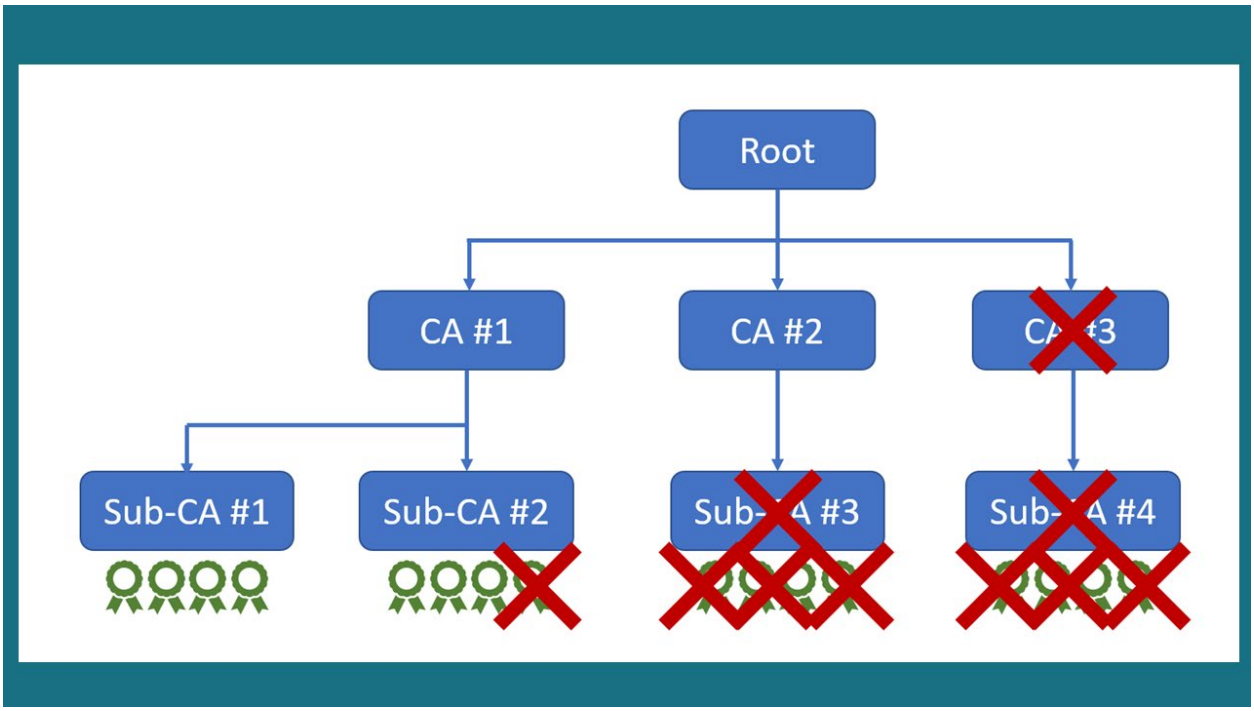
Let's illustrate PKI. In the below diagram, we can see that every CA provides rights to sub-CAs (Subordinate CAs) to sign digital certificates for devices. And these digital certificates stay at the end of the hierarchy. They are accepted by the devices and authorized by the sub-CA that signs and generates them. They are also called as device certificates.



Example of Public Key Infrastructure

Moreover, the sub-CAs that generate the device certificates have their certificate, which is authorized by the digital signature of the Certificate Authority (CA) that sits above them. Ultimately, PKI is at the top which is the foundation and root of this PKI environment hierarchy.

Apart from this, some of the reasons for the PKI environment to be arranged in hierarchies is because it allows to revoke or deny access to selected levels if a leak or compromise of a private key happens. Below is an example of the same.



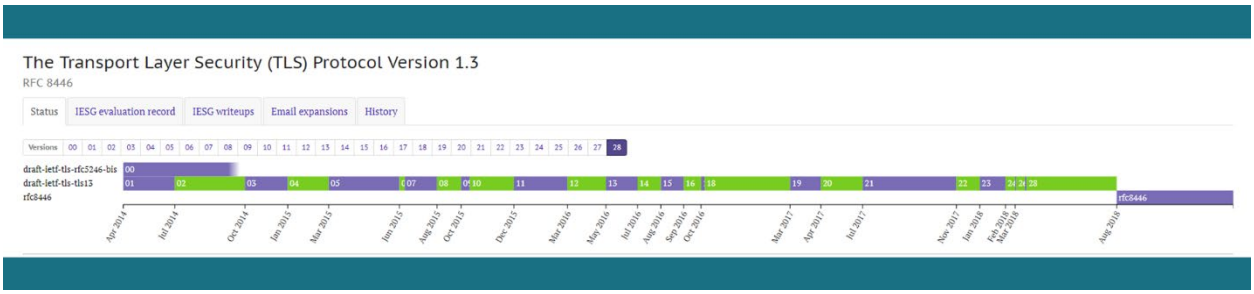
Example of Certificate Revocation by Public Key Infrastructure

By looking at the above image, it's not hard to understand why PKI is implemented through tree type hierarchies. The main benefit of PKI through the hierarchy is that it allows the owner of the environment to control the compromised event. It is also one of the reasons why device certificates are not issued directly through the root CA but through a sub-CAs that are below the root, because if something goes wrong, the entire PKI and all the deployed devices of that field will need to be revoked.

Lastly, one of the main benefits is that a single sub-CA is capable and is generating more than millions of device certificates used to authenticate millions of devices with only one single public key of sub-CA.

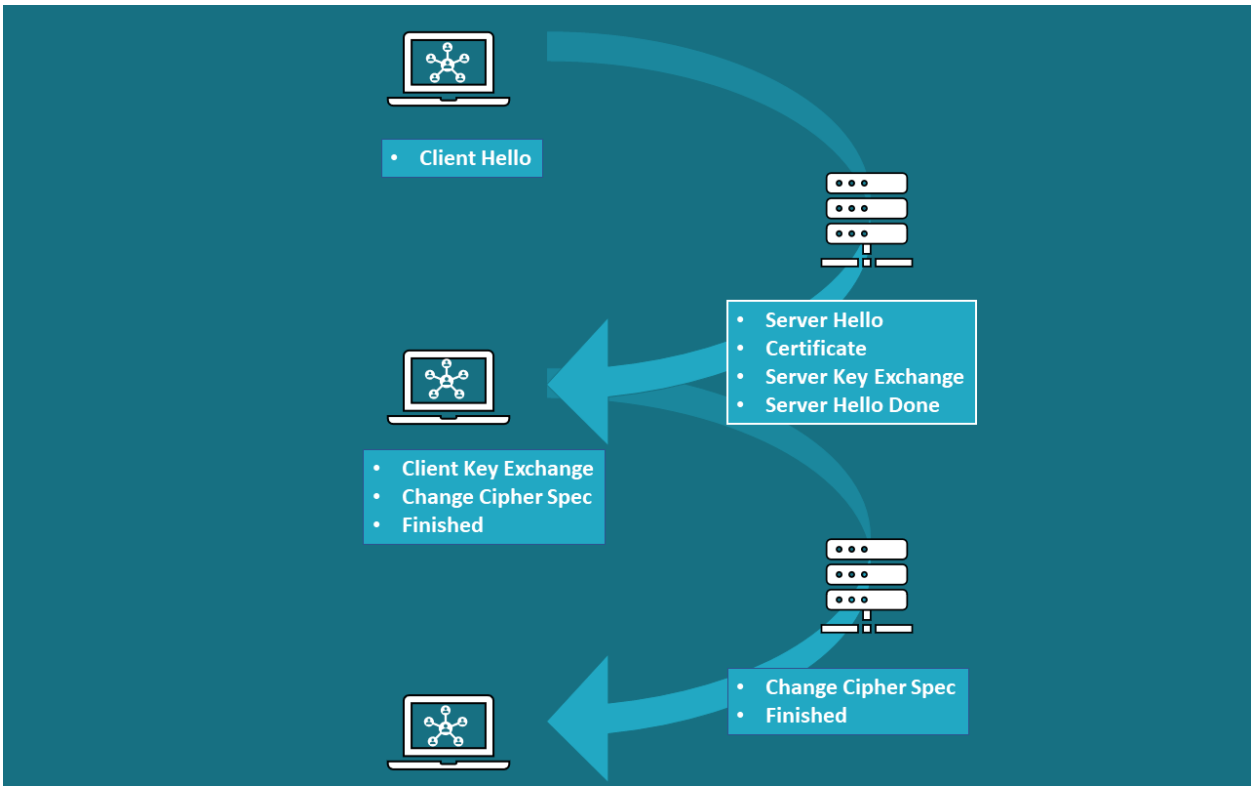
TLS 1.3: Faster and More Secure

TLS 1.3 (RFC 8446), the seventh iteration of the SSL/TLS protocol has taken around 28 drafts and ten years to get defined after TLS 1.2. Also, it came across problems with middleboxes and many others. But eventually, it got released and arrived with some significant improvements to provide unparalleled performance and privacy compared to its previous versions.



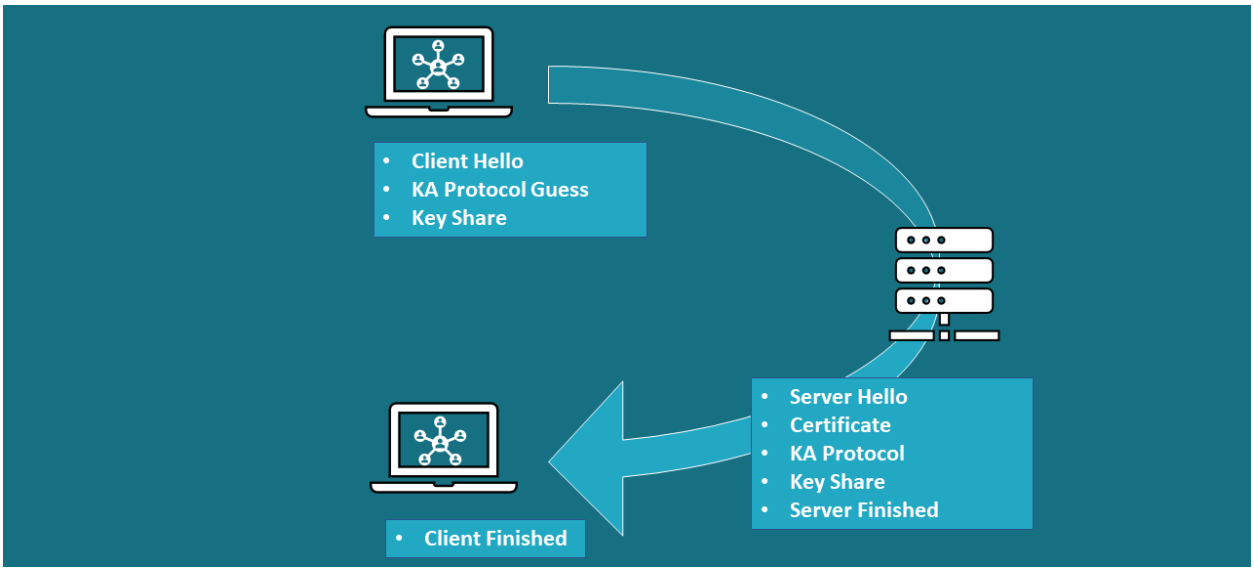
Also, from the very first version of TLS 1.3, which was released on April 17, 2014, all the way till its 28th and the final version, all the drafts were tested and reviewed by tech giants like Google, Mozilla, Cloudflare and many more while reporting the faced issues. For example, in February 2017, a proxy issue was faced. This forced Google to back off from supporting TLS 1.3 entirely for a while, which caused a further delay.

TLS 1.3 made its major improvement by making changes in the Handshake protocol. In TLS 1.2, it went through two round trips for the completion of Handshake.



TLS 1.2 Two Roundtrip Handshake

In TLS 1.3, it takes only one round-trip while making its connection site faster. Because the negotiations taking place between the client and the server have been reduced to two from four, Key exchanges and digital signature scheme through extension are not required anymore. As everything happens in milliseconds, it does not get noticed so quickly, but it does make a difference.



TLS 1.3 Single Roundtrip Handshake

Let's see the TLS 1.3 protocol steps in detail. The significant difference is that the TLS 1.3 handshake protocol involves one round trip compared to TLS 1.2, which has three, eventually resulting in reduced latency.

Sr No.	Message	Action	Note
1	✓ ClientHello ✓ Supported Cipher Suites ✓ Guessing of Key Agreement Protocol ✓ Key Share	Here, it initiates with the message "ClientHello," but the difference is that the client also sends the list of supported cipher suites while guessing which key agreement protocol will be selected by the server. Lastly, the client also sends its key share regarding that agreement protocol.	The first step of TLS 1.3 is quite similar to the TLS 1.2 handshake.
2	✓ ServerHello ✓ Key Agreement Protocol ✓ Key Share ✓ Server Finished	The server replies with its chosen vital agreement protocol, where "ServerHello" also includes the server's key share as its certificate and lastly "ServerFinished" message.	If you notice the difference here, you might have seen that the "Server Finished" message is sent in the 2nd step itself, which saves time as well as one round trip.
3	✓ Checks Certificate ✓ Generate Keys ✓ Client Finished	Finally, the client verifies the server certificate, generates its keys because of the server's key share and sends the message "Client Finished," which means data encryption can be started.	-------

Moreover, TLS 1.3 went further and made its advancement to enable 0-RTT handshake even before the client and server meet due to which it takes zero round trips to do the handshake, resulting in the improvement of latency.

You can say, it is more like a milestone, as due to *accomplishing 0-RTT Resumption*, the client can connect with the server, even before TLS 1.3 permits a zero-round trip handshake. It is typically accomplished by storing secret information such as Session ID or Session Tickets of previous sessions and using them for future use whenever both parties connect.

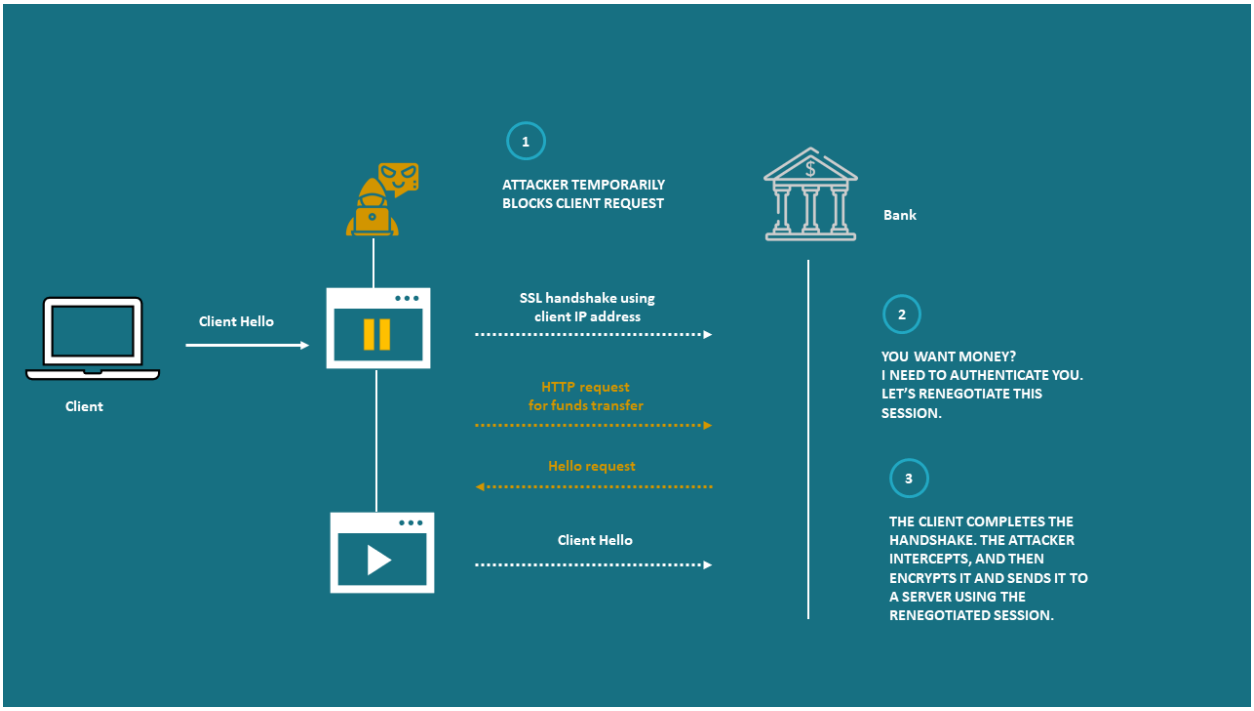
Benefits of TLS 1.3 over TLS 1.2

TLS 1.2 provides some extensive improvements over TLS 1.2. Hence, optional parts that caused the chances of vulnerability have been removed and provided support for stronger ciphers, which is required to implement Perfect Forward Secrecy (PFS) and other like short handshake process.

Security

Granted, it's possible to deploy TLS 1.2 securely, but the optional parts of the protocols and some outdated ciphers have been exploited by several high-profile vulnerabilities. Here, TLS 1.3 helps by removing those problematic options and provides support to algorithms which is not known to any vulnerabilities till date.

The renegotiation attack is an example of such a threat which is not possible in TLS 1.3.



Renegotiation Attack

Likewise, IETF has chosen to remove all ciphers which do not support PFS (Perfect Forward Secrecy) from TLS connection such as AES-CBC, DES, RC4 and some other less commonly used ciphers. IETF also removed the capability of performing "renegotiation," which allowed the client and server to generate new keys, negotiate new parameters, etc. which is already there with the TLS connection. Eventually, eliminating the chances of a renegotiation attack.

Privacy
PFS is enabled by default in TLS 1.3. Due to this, an additional layer of confidentiality is added to an encrypted session, which ensures that the traffic can only be decrypted by the two endpoints. Additionally, even if someone tries to record an encrypted session and later get access to the private key of the server, PFS will not allow using that key for decrypting the session.

Performance
As already discussed, with TLS 1.3, the entire round trip is eliminated from the connection while establishing handshake, resulting in half encryption latency. Additionally, it's possible to send data to the server on the first message itself, if you access a previously visited website, which is known as 0-RTT (Zero Round Trip Time).

TLS 1.3 vs. TLS 1.2

Some of the substantial differences between the two are as below:

Removed Vulnerable Algorithms and Ciphers

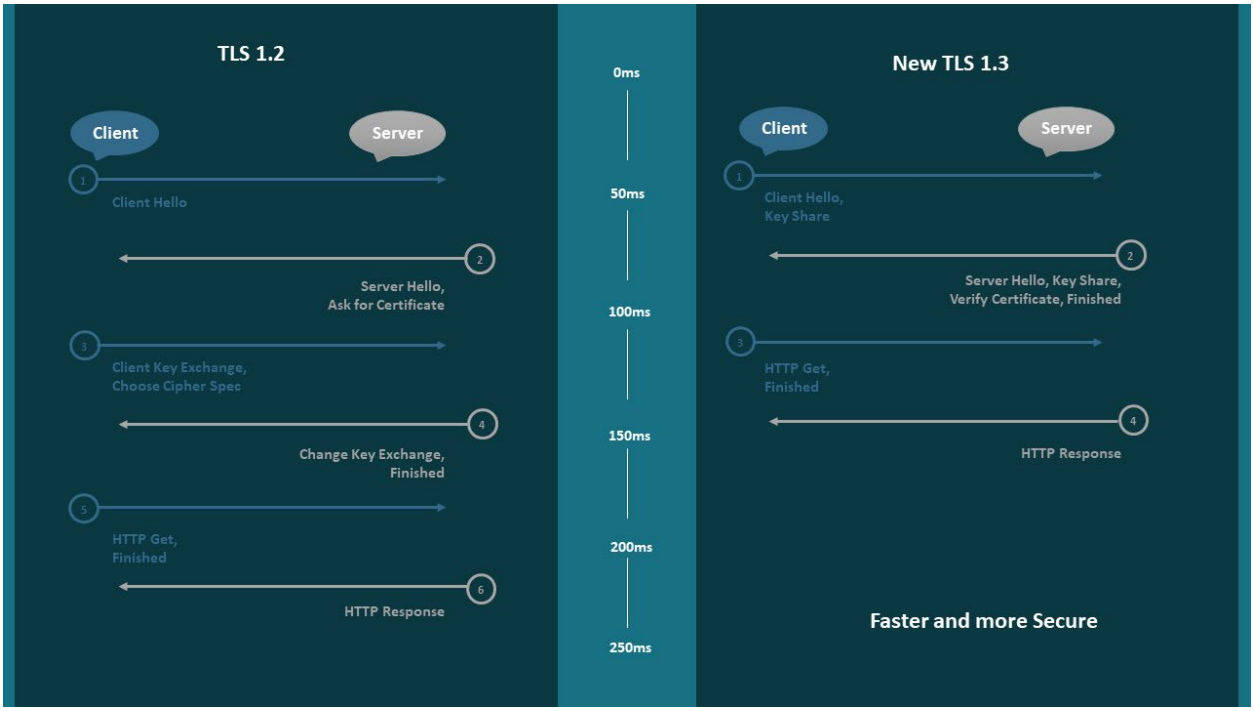
It removed support for ciphers and algorithms that are both theoretically and practically vulnerable to attacks:

- RSA Key Exchange
- RC4 Stream Cipher
- CBC (Block) Mode Ciphers
- MD5 Algorithm
- SHA-1 Hash Function
- Many short-lived Diffie-Hellman groups
- DES
- 3-DES
- Export Strength Ciphers

Removed RSA key exchange while making Perfect Forward Secrecy (PFS) mandatory

RSA and Diffie-Hellman, two popular mechanisms were used to exchange the secure session key at the time of HTTPS connection next to the SSL handshake. Now RSA, along with all static (non-Forward Secret) has been removed as it has specific problems like Oracle Padding Attacks.

Moreover, RSA doesn't offer an ephemeral key mode, which is mandatory for PFS. And the reason to make PFS compulsory is that without that the private key can get compromised and the encrypted conversation can be decrypted, and if there's perfect forward secrecy, it can provide security against it. Eventually, reducing the time by eliminating one entire round-trip and giving better performance of the website.

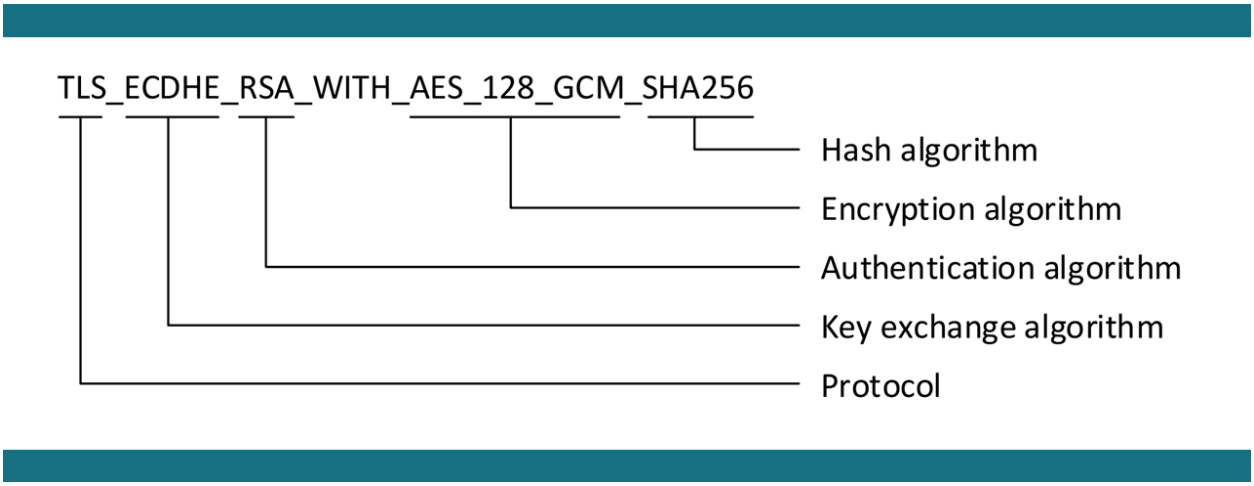


Side-by-Side TLS 1.2 and TLS 1.3 Handshake Comparison

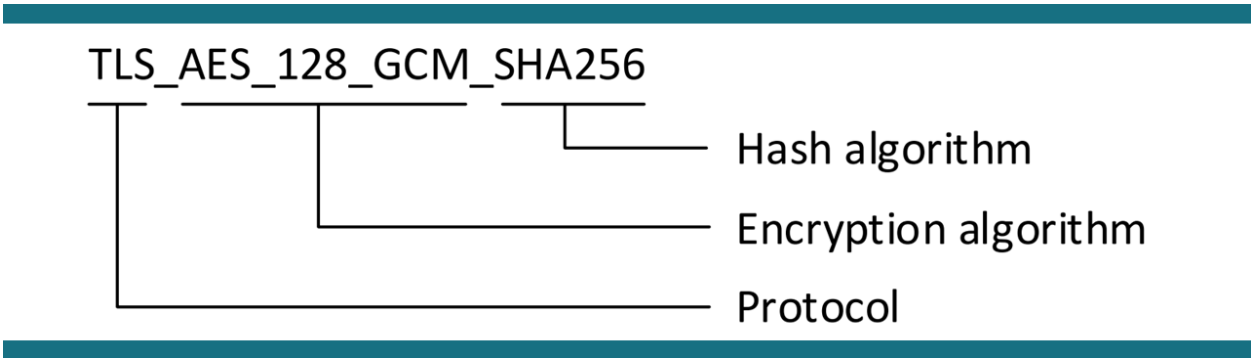
Easier Cipher Suites

As the negotiations have been eliminated from the handshake process, the size of the cipher suites has also got smaller.

TLS 1.2 and its prior versions used cipher suites, which included four ciphers. For example:



Now, in TLS 1.3, cipher suites do not include the key exchange and signature algorithms. Now, it's only bulk cipher and the hashing algorithm. For example:



The cipher suites are registered and maintained in the TLS Cipher Suites Registry by IANA, giving every cipher suite its unique number for identification. So, the cipher suites defined for TLS 1.3 cannot be used with TLS 1.2 and vice-versa, even if they use the same cipher suites.

Moreover, the defined encryption algorithm of the TLS 1.3 cipher suite must be an AEAD (Authenticated Encryption with Additional Data) algorithm, as it provides both confidentiality and message authentication in a single crypto algorithm. Here, the main concept of the integrity and encryption function has been changed with the AEAD algorithm.

Due to the introduction of AEAD and reduction of the supported cipher, it will not be possible to send unencrypted data through TLS 1.3, which was an option in a TLS 1.2 by using NULL encryption cipher suites.

Lastly, the recommended cipher suite list has also shrunk down significantly.

TLS 1.2 Cipher Suite List:

- ✓ TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
- ✓ TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
- ✓ TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
- ✓ TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
- ✓ TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
- ✓ TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
- ✓ TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
- ✓ TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
- ✓ TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
- ✓ TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384

- ✓ TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
- ✓ TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
- ✓ TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
- ✓ TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
- ✓ TLS_DHE_RSA_WITH_AES_128_CBC_SHA
- ✓ TLS_DHE_RSA_WITH_AES_256_CBC_SHA
- ✓ TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
- ✓ TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
- ✓ TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256
- ✓ TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305
- ✓ TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256
- ✓ TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305

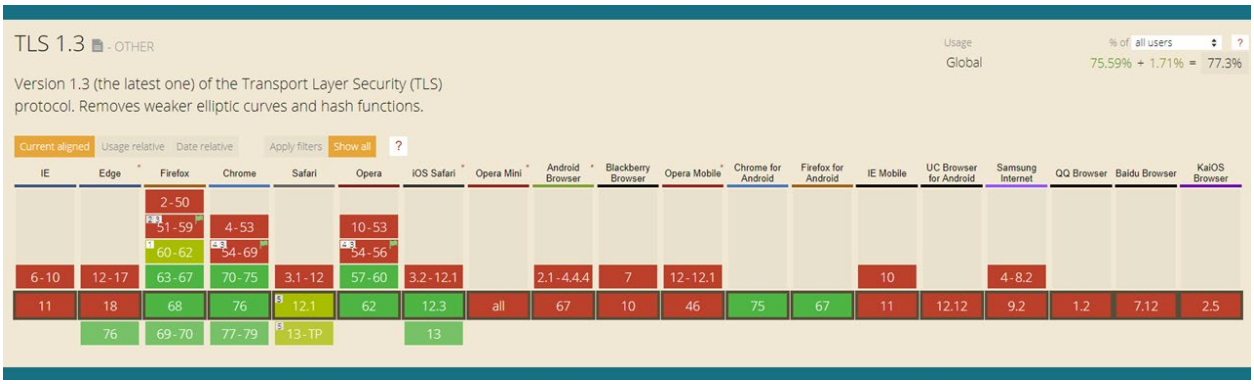
TLS 1.3 Cipher Suite List:

- ✓ TLS_AES_256_GCM_SHA384
- ✓ TLS_CHACHA20_POLY1305_SHA256
- ✓ TLS_AES_128_GCM_SHA256
- ✓ TLS_AES_128_CCM_8_SHA256
- ✓ TLS_AES_128_CCM_SHA256

Browsers Supporting TLS 1.3

Mostly, all the popular web browsers support TLS 1.3:

- ✓ Google Chrome Ver. 67+
- ✓ Mozilla Firefox Ver. 61+
- ✓ Apple - Mac OS 10.3 and iOS 11



Client-Side TLS 1.3 Support Chart

Moreover, Chromium-based Edge browsers support it, whereas Microsoft is a bit lagging with their operating system and browsers.

How to Upgrade TLS 1.3 on Server?

TLS 1.3 update is similar to upgrading any software library. Simply, update SSL/TLS library to one of the below versions:

- ✓ GnuTLS 3.5x
- ✓ Google's Boring SSL (Latest)
- ✓ FaceBook's Fizz (Latest)
- ✓ OpenSSL 1.1.1

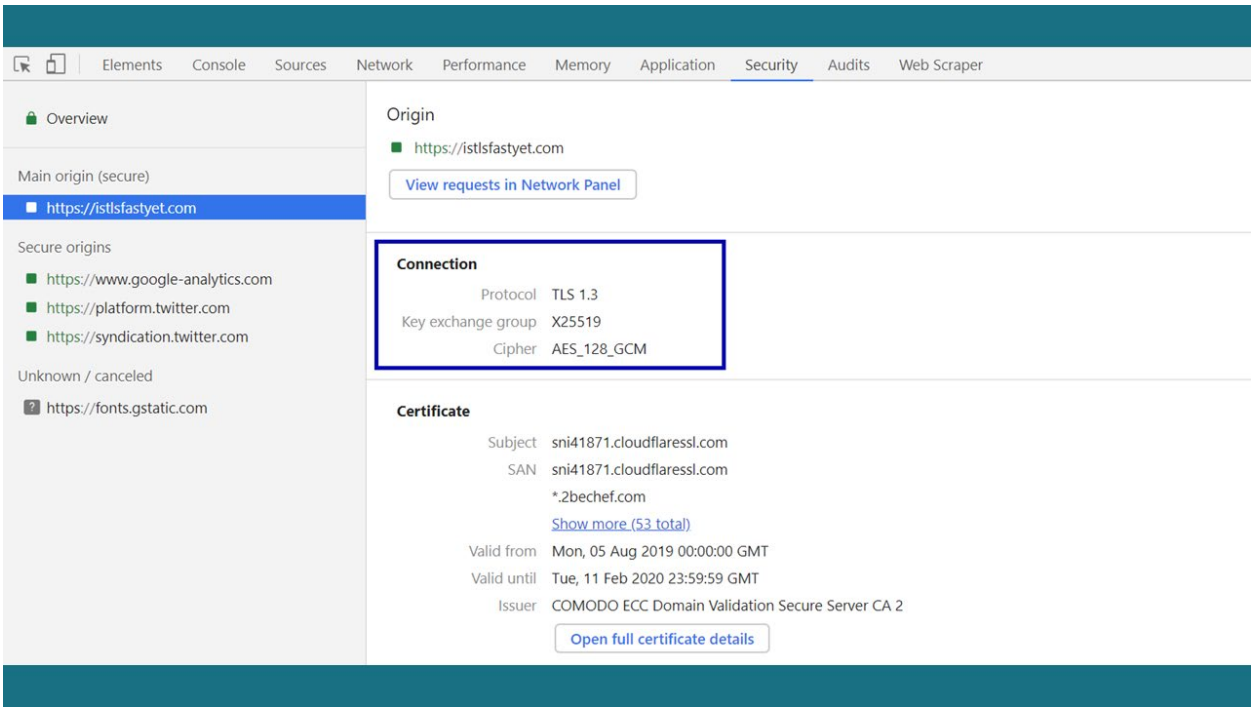
Moreover, several hosting and service providers such as Cloudflare, Google, Akamai and FaceBook support TLS 1.3 connections. But the possibility of configuration depends on the server as most of the work is done during software library upgrade to a TLS 1.3 supported version.

How to Enable TLS 1.3 in Different Web Browsers?

Different web browsers require different steps.

Enabling TLS 1.3 in Google Chrome

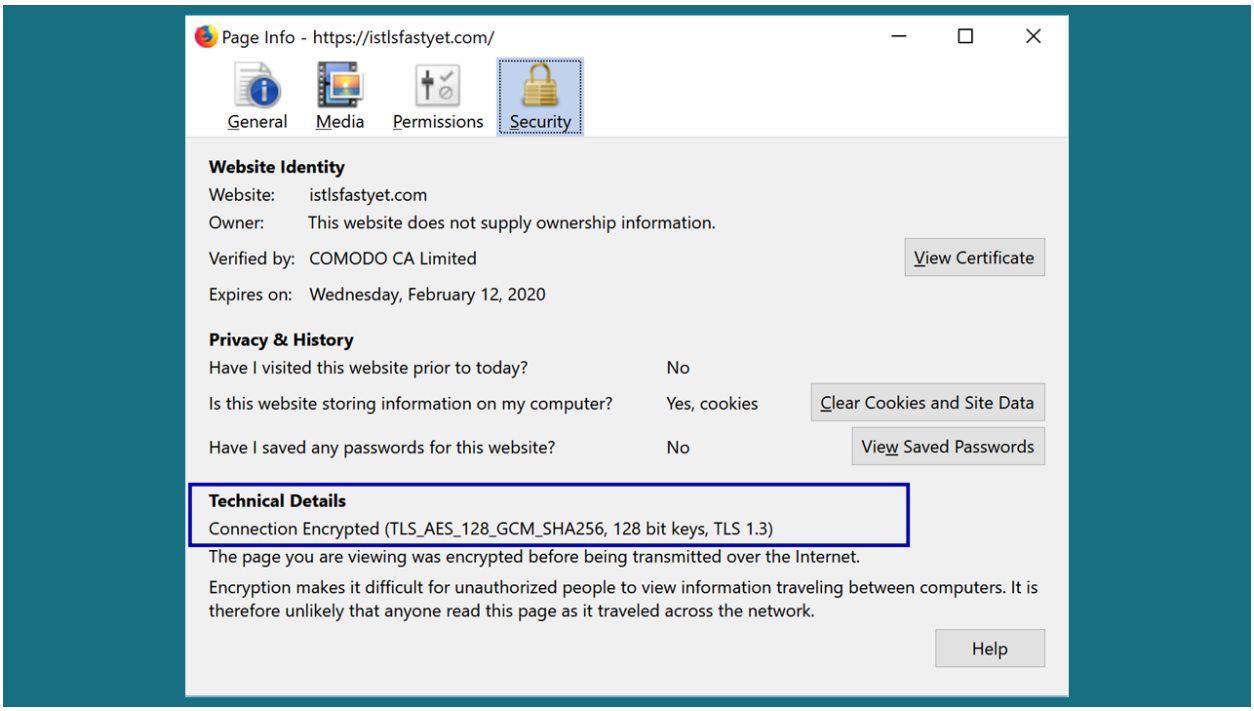
- ✓ Open Google Chrome
- ✓ In the address bar, type “chrome://flags/” and press Enter.
- ✓ Go to TLS 1.3 and select the option Enable TLS 1.3 (Draft 23)
- ✓ Now, go to https://istlsfastyet.com/ and press F12 and go to the tab “Security”
- ✓ Reload the website and under the “Main origin,” click on the listed link



Now, you will be able to see that your connection to the website is protected through TLS 1.3.

Enabling TLS 1.3 in Mozilla Firefox

- ✓ Open Firefox and type “about:config” in the address bar and hit Enter
- ✓ Search for tls.version.max
- ✓ Double click on it and change the value to 4
- ✓ Restart Firefox and go to https://istlsfastyet.com/
- ✓ In the URL bar, click on the padlock icon, go to Connection and under that, click on More Information
- ✓ A Certificate Details window will open and in that, at the bottom under technical details, you will find that the website is protected with the TLS 1.3 protocol.



TLS 1.3 Errors

Browser Error
Users may see an error while opening your website
ERR_SSL_VERSION_INTERFERENCE

It means TLS version is not available mutually between the client and the server for the connection. It generally happens when there's a mismatch in TLS 1.3 versions supported by the web browser and the web server.

Note: Due to these types of situations, it's essential to support TLS 1.2.

TLS 1.3 Older Version Error
Generally, this problem occurs when the older version of TLS 1.3 draft process is used, which was available before the protocol was finalized. To avoid, be assured you and your website visitors are using the latest TLS 1.3 version with the respective browsers, software libraries and OSs. Trying to connect via an outdated draft version of TLS 1.3 may cause errors.

Is it a Right Time to Start using TLS 1.3?
It has almost been a year since TLS 1.3 has been officially released, yet it has not been adopted as it has to be. But, due to the compulsion of SSL/TLS and HTTPS and the users being more aware of cybersecurity, maybe it will be used at large in the next two to three years.

Trust Hierarchy

For certificates to be effective, their users must trust them. There have been cases where users were not able to trust the issuer of a certificate. For example, a user hearing about a certificate authority for the first time and feels uncomfortable accepting a certificate from that issuer. So, it's an obvious question, how or who decides whether the CA is trustworthy.

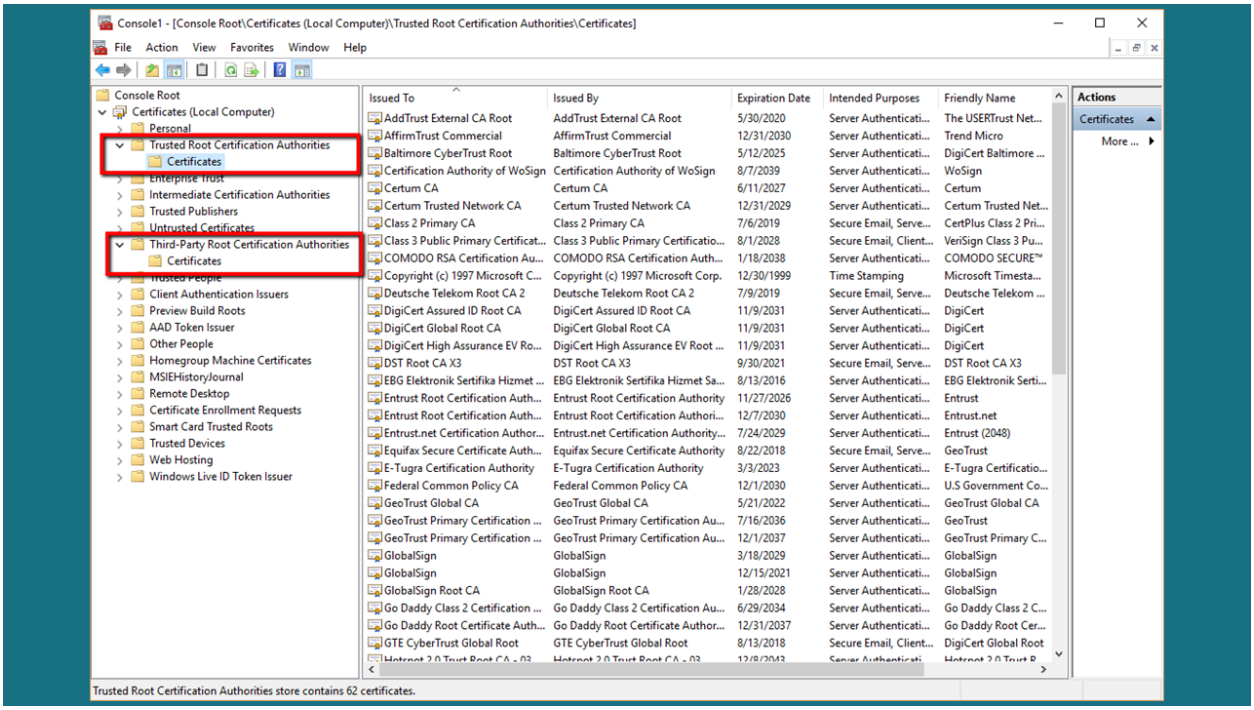
Who decides whether a CA is Trustworthy?

Authorized CA 'membership' programs are operated by operating systems, browsers and mobile devices, where the CA must meet the detailed criteria for being a member. Once the CAs are accepted, they can issue SSL Certificates that are trusted by browsers, devices and people relying on it.

However, there are a limited number of authorized CAs. Whether they are private companies or government entities, the longer the CA is active, the more the browsers and devices will trust the certificates issued by that CA. Likewise, to be trusted, certificates must be able to provide backward compatibility with older versions of browsers and specifically for mobile devices. This is also known as ubiquity, one of the important features of a CA. All these happen in a hierarchical manner, which is also known as Trust Hierarchy of a CA.

Hierarchy of Trust

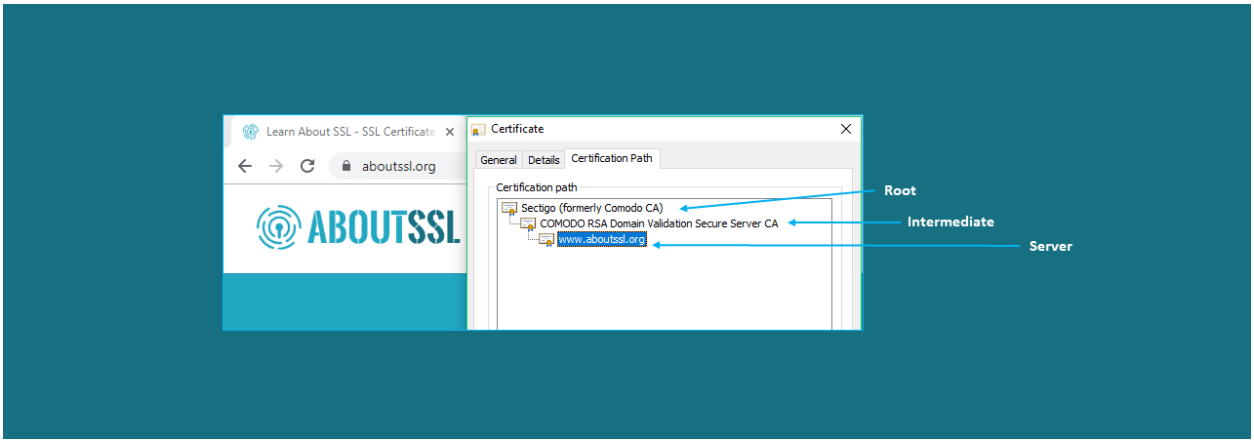
Browsers, operating systems, devices and even mobile carrier operate a root store which is a database of all the approved CAs that comes pre-installed in them. Also, CA is trusted if their Root Certificate is accepted by that root store.



Certificate Store in Microsoft Windows Operating System

While issuing their Intermediate Root Certificates and End Entity Digital Certificates, they use these pre-installed Root Certificates. The process includes certificate request, further validating the applications, issuance of the certificate and finally publishing the current validity status of that issued certificate so that the user who relies on it can know that certificate is valid.

Generally, CAs create several Intermediate CA (ICA) Root Certificates that are used to issue end-entity certificates, for example, SSL Certificates, referred to as Trust Hierarchy and it looks like below:



Certificate Authorities (CAs)

A Certification Authority or Certificate Authority (CA), is one of the trusted entities that issues security certificates. These certificates help in verifying the ownership of public keys used to secure communication on the internet. It is a part of the PKI (Public Key Infrastructure) with the Registration Authority (RA) who helps in verifying the information provided by the requester of a security certificate. And, once the requested information is verified, CA will issue a certificate.

In other words, a CA is a trusted third-party entity that offers security certificates in the X.509 format, standard to organizations that wish to assure users that they provide secure authentication and connection. These security certificates, generally known as SSL/TLS Certificates provided by CAs, which helps in building trust between the users and the providers.

The business of Certificate Authority is fragmented worldwide with providers dominating their home market. But the market for globally trusted SSL/TLS Certificate who provides it legally binding digital signatures and links to regulations, local law and accreditation schemes of certificate authorities are still limited with the handful of multinational CAs like Sectigo, DigiCert Group, GoDaddy Group, GlobalSign and further their resellers like TheSSLStore, RapidSSLOnline, CheapSSLSecurity.

Root Programs

Some major softwares come equipped with a list of CAs (Certificate Authorities) that are by default trusted. Due to this, it becomes easier for users to validate certificates and easier for organizations or people who are requesting a certificate to know which certificate authorities are trustworthy and can issue a widely trusted certificate. And it's one of the important things every site administrator will look for while getting a certificate, as they would like to go for a certificate which is trusted by nearly all the website visitors. For all these, documents and guidelines are maintained under the regulations and restrictions made by the CA/B Forum's Baseline Requirements, known as Root Certificate Programs (RCP) and they have the responsibility to tell what procedures are required by a company to get the Root Certificate to be included in the browser.

In other words, policies and processes used by a software provider to decide which Certificate Authority's Root Certificate should be trusted by the software are known as Root Programs or Root Certificate Programs. Lastly, to maintain its trust, it processes through many 3rd party auditing.
Some of the globally known Root Programs are:

- ✓ Apple Root Program
- ✓ Oracle Java Root Program
- ✓ Mozilla Root Program
- ✓ Microsoft Root Program

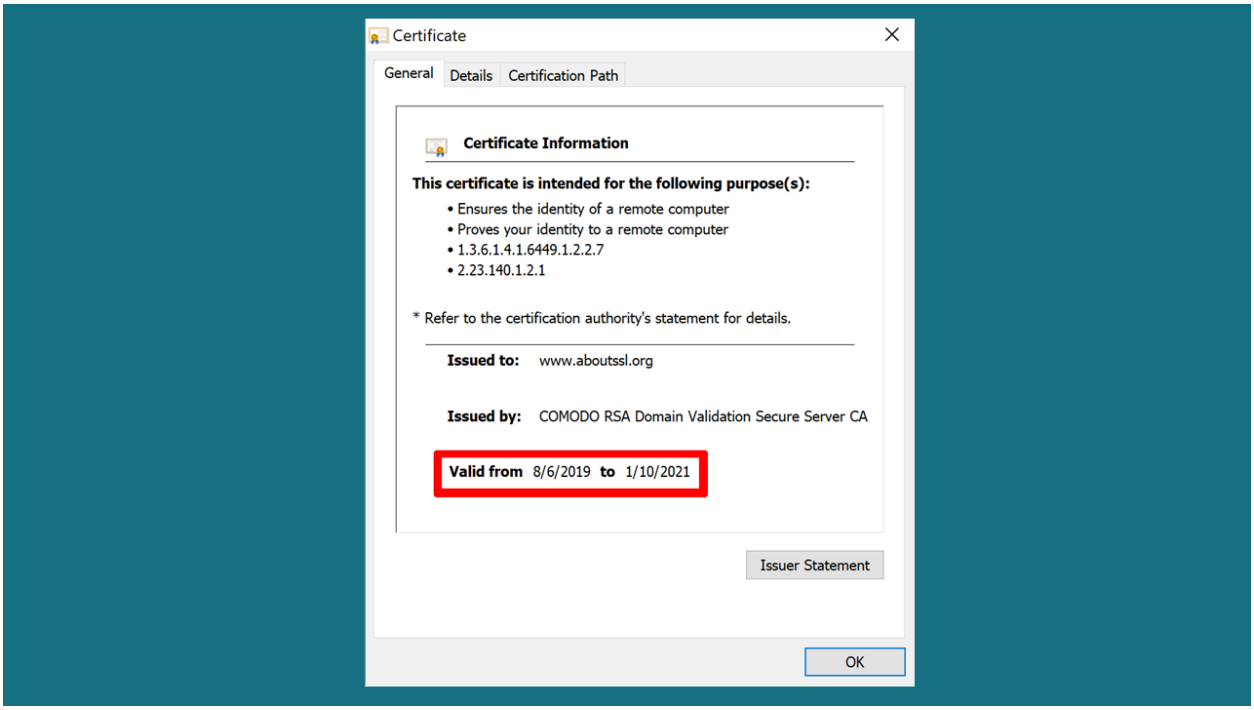
Roots/Intermediates

Security provided by SSL certificates is based on a Chain of Trust that originates from the Root Certificate of Certificate Authorities like Sectigo, DigiCert, GeoTrust, etc to the user's certificate. And further, these SSL certificates are accepted by popular web browsers that contain the validated digital signature of the CA. Though the CA's identity is built by adding the root certificates in the web browsers and without that, no browser would know whether to accept an SSL Certificate issued by a CA.

Furthermore, the Certificate Authorities are very strict with their guidelines and to guard their Root Certificates from getting compromised, they use an Intermediate Certificate which helps in issuing the user's end-entity or leaf certificate.

Let's understand what Roots and Intermediate Certificate are.

Root Certificate is a public key certificate, which helps in identifying a root certificate authority. These root certificates are self-signed while forming the basis of X.509 based PKI (Public Key Infrastructure), used to issue other certificates. Additionally, the lifespan of these root certificates is more than the leaf SSL certificates, that are two years and one CA can have many root certificates.



Comodo's Root Certificate Example

As mentioned earlier, the Intermediate Certificate is used to provide security to the Root Certificate. Moreover, Intermediate certificate also works as a trusted root for issuing a leaf or

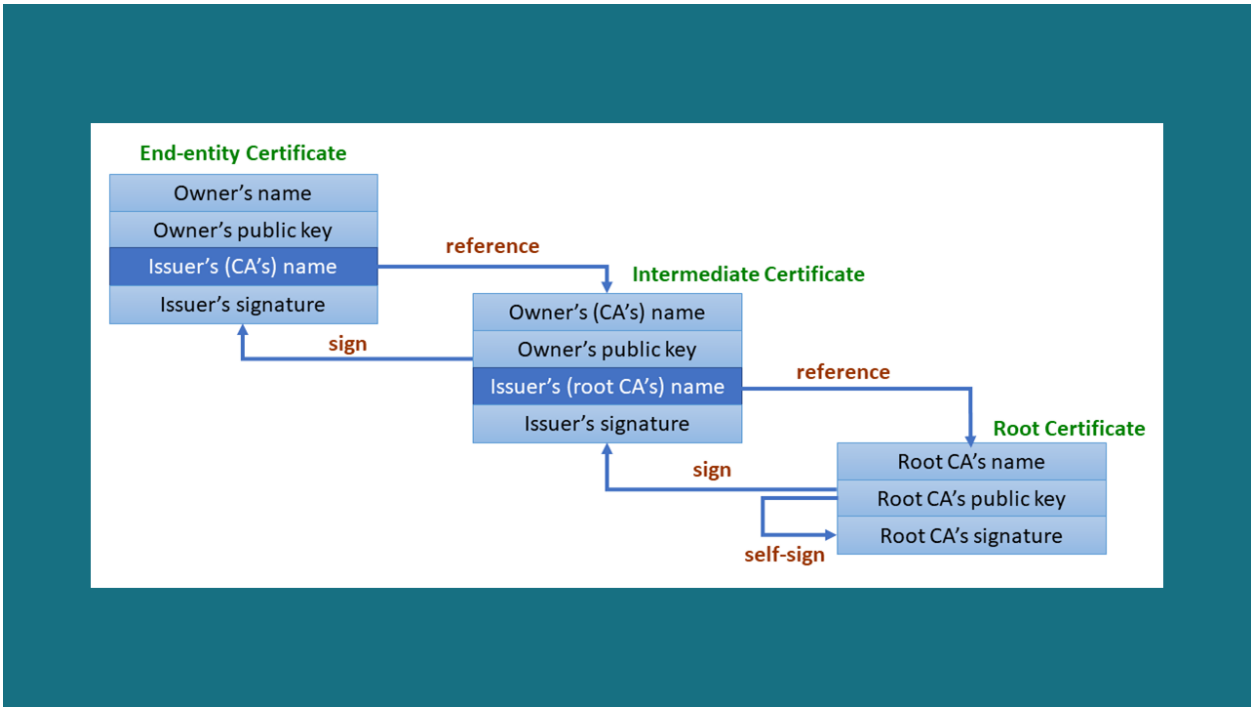
end-entity server certificate, which results in a chain of certificates that starts with the trusted root CA, then the intermediate and lastly the leaf certificate, which is an SSL certificate issued to users.

Leaf Certificate

A leaf certificate, also known as the end-entity certificate, is the last non-CA certificate of the chain which consists of a public key which is used by the users. These are the end-users' SSL Certificates which is not directly issued by the Certificate Authorities from their roots, as those roots are much valuable and a lot of risks are involved around it.

Certificate Chain / Chain of Trust

A chain of trust is established by validating the component of hardware and software from the end entity to the root certificate. The main reason behind establishing it is to ensure that the trusted software and hardware keep getting used while keeping the flexibility.



Chain of Trust

Moreover, Digital Certificates are verified via a chain of trust which is a list of certificates in an order. This contains End-Entity Certificate (Leaf Certificate), Intermediate Certificate and Root Certificate, where the root certificate authority (CA) is the trust anchor for the digital certificates.

TYPES OF SSL CERTIFICATES

SSL/TLS Certificates are available as per the requirements, as different websites serve different purposes. For example, a blogger who writes posts and does not ask any sensitive information of the visitors will need a different SSL Certificate compared to globally known online shopping portals like Amazon.com. Hence, Certificate Authorities offer different types of SSL/TLS Certificates.

Additionally, SSL/TLS Certificates are categorized based on two characteristics:

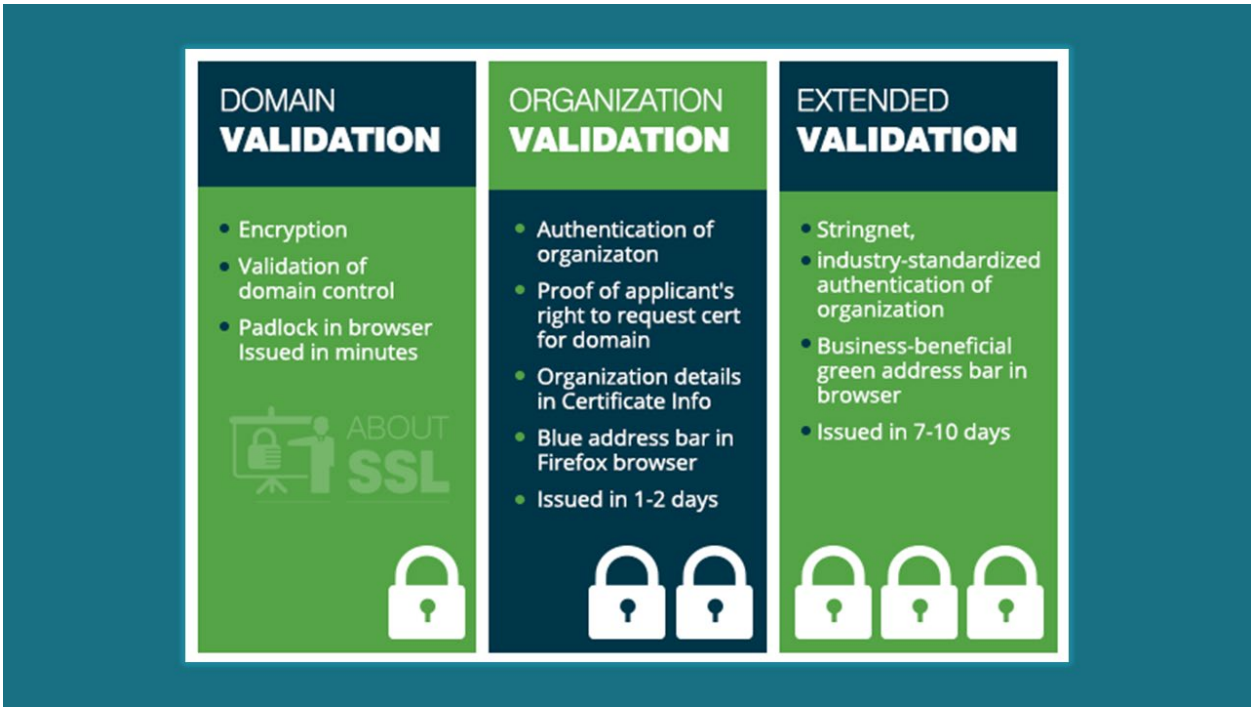
- ✓ Validation Level
- ✓ Certificate Functionality

Let's look at each of them in detail.

What are the Different Validation Levels of SSL Certificates??

When it comes to validation of SSL/TLS Certificates, there are three different levels. Below are the three different types of validation:

1. Domain Validation (DV)
2. Organization Validation (OV)
3. Extended Validation (EV)



1. **Domain Validation (DV)** – Domain Validated SSL certificates are entry level certificates and majority of the active SSL/TLS Certificates we see on the internet are Domain Validated. The main reason is apparent; it's easier to get these SSL certificates. All a person needs to do is to prove ownership of their domain. A DV SSL/TLS Certificate will be issued in just minutes, and it's very affordable too. These certificates are popular among small websites and bloggers who are just looking for simple encryption. Almost anyone can get this certificate.

2. **Organization Validation (OV)** – OV was the first SSL certificate that came into existence. Later, DV was created to expand greater access towards encryption, and EV was created to offer the highest degree of authentication. But the very first was OV. To get an Organization Validated SSL/TLS Certificate, it's mandatory for an organization to go through a light business vetting process, in return, they get the certificate which displays the details of the verified business. Before issuing an OV SSL certificate, the CA will verify the organization credentials of the applicant.

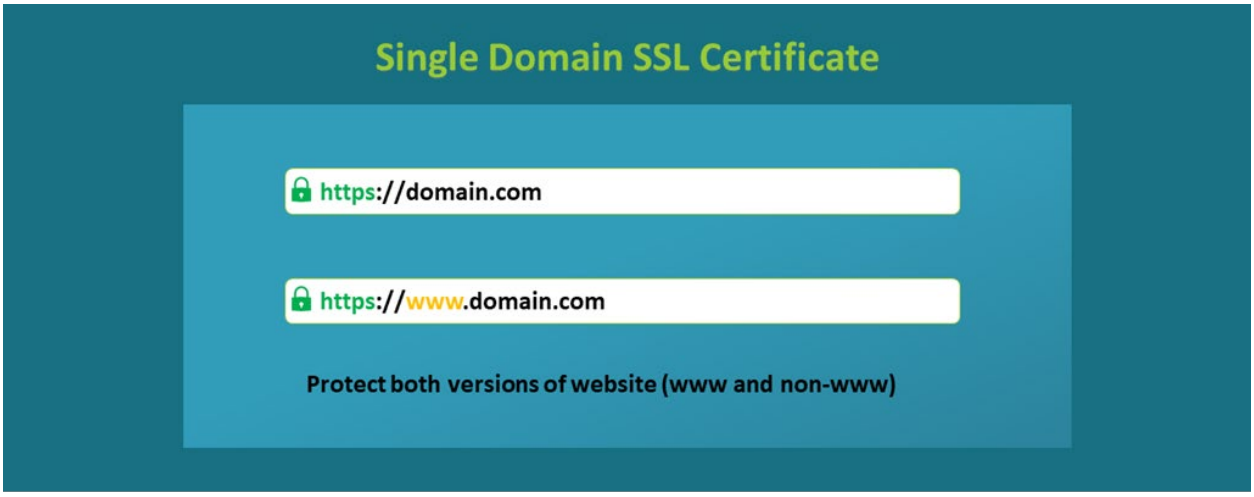
3. **Extended Validation (EV)** – EV SSL/TLS Certificates offer the highest level of authentication. The vetting process of an EV SSL certificate is also very stringent when compared to DV and OV SSL certificates. It verifies several things regarding your business, such as Physical Address, Telephone Verification, and Operational Existence of your business, which can take anywhere from 1 to 5 days.

Different Types of SSL Certificates

There are four different types of SSL/TLS Certificates.

- ✓ Single Domain
- ✓ Multi-Domain
- ✓ Wildcard
- ✓ Multi-Domain Wildcard

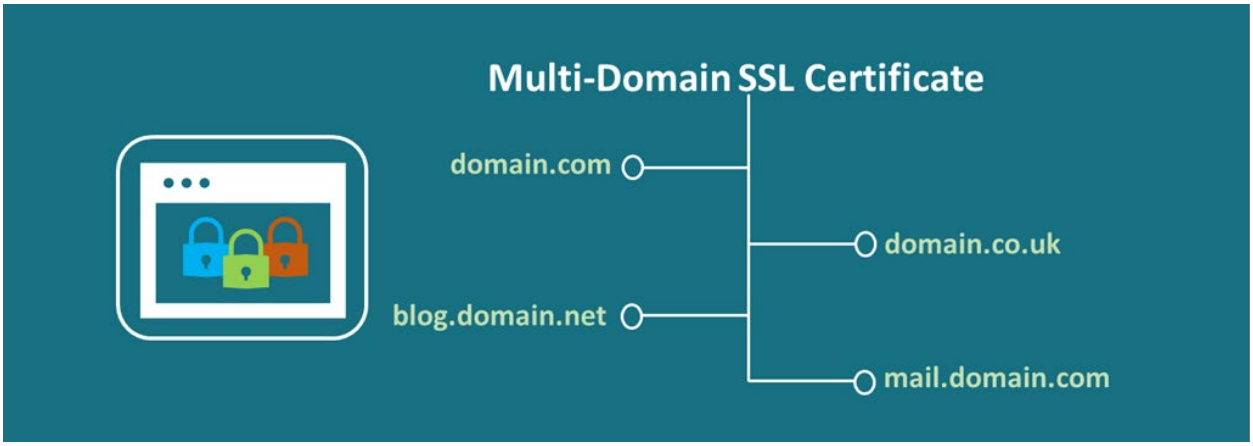
Single Domain SSL



A Single domain SSL/TLS Certificate protects only one specific domain. It's one of the most popular and commonly used certificates that protects the www and non-www versions of the domain. It cannot be used for any other domain as well as subdomains of the main one which is covered.

For example, a single domain SSL certificate can protect www.My-New-Domain.com and My-New-Domain.com but it cannot protect blog.My-New-Domain.com or mail.My-New-Domain.com

Multi-Domain/SAN SSL



When it comes to SSL/TLS Certificates, administrative burden and cost of purchasing them for multiple domains are one of the biggest issues faced by many organizations and companies. Fortunately, Certificate Authorities do offer a solution for these issues through Multi-Domain SSL certificates.

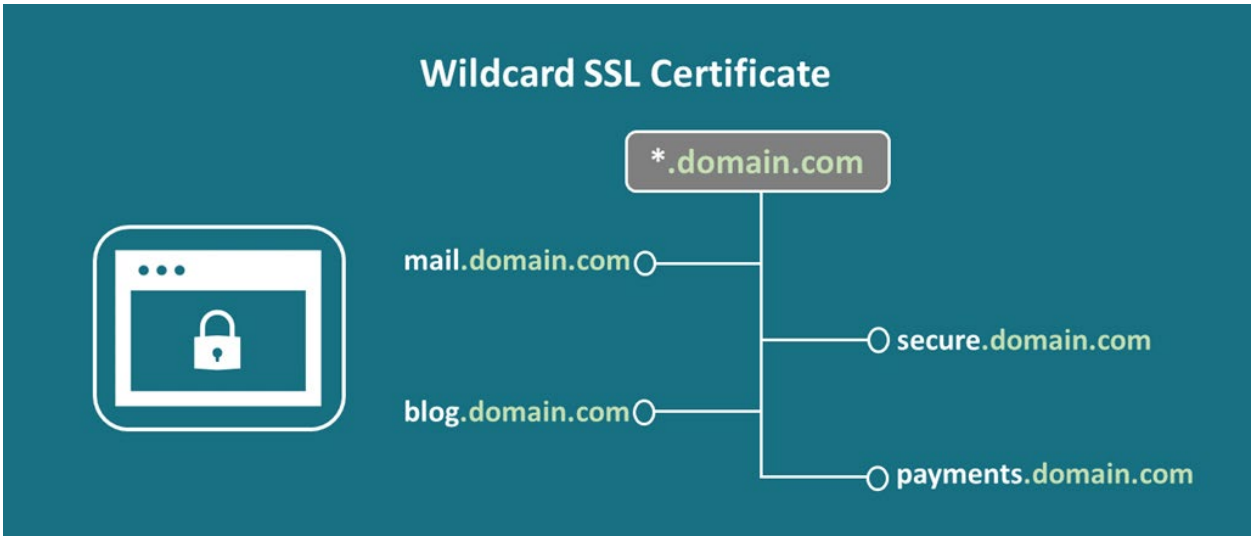
Multi-Domain/SAN certificates also called as UCC (Unified Communications Certificates) are certificates that can encrypt multiple domains. These certificates will help save a lot of time and money as a single certificate is enough to secure up to 250 domains and subdomains. Multi-Domain certificates come with 2 to 4 SANs and based on your needs, you can purchase additional SANs. If you are an organization that has multiple domains and subdomains, Multi-Domain/SAN certificates are for you.

For example, a multi-domain SSL certificate can secure the following domains

- ✓ www.My-New-Domain.com
- ✓ www.My-New-Domain2.com
- ✓ secure.My-New-Domain.com
- ✓ www.My-New-Domain.org
- ✓ mail.My-New-Domain.com.net
- ✓ dev.My-New-Domain2.org

Wildcard SSL

Do you have one main domain with multiple sub-domains? If yes, you do not have to purchase separate SANs to encrypt the main domain and all its sub-domains? Certificate Authorities offer Wildcard SSL/TLS Certificates that can help you *protect an unlimited number of sub-domains*.



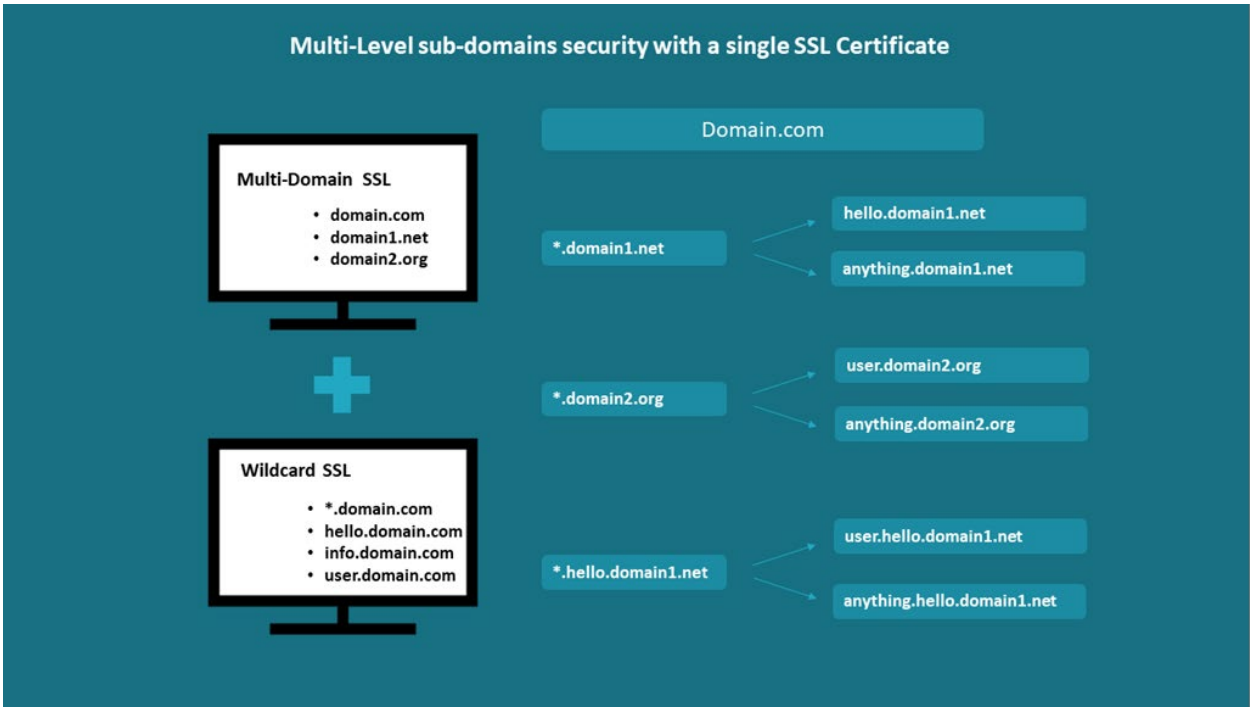
Wildcard SSL is one of the most versatile SSL Certificates that can provide encryption for an unlimited number of subdomains on one single certificate. You're allowed to add an unlimited number of subdomains later on, as long as you're reissuing your purchased certificate.

For example, a wildcard SSL certificate you purchase for www.My-New-Domain.com, will allow you to secure its subdomains like the following.

- ✓ mail.My-New-Domain.com
- ✓ forum.My-New-Domain.com
- ✓ blog.My-New-Domain.com
- ✓ and many more

Moreover, EV Wildcard SSL/TLS Certificates are not available, so there's no other way around other than settling for an OV. If you're looking to encrypt sub-domains of multiple domains, then you will need to purchase multiple Wildcards for those different domains.

Multi-Domain Wildcard SSL



Finally, here is the jack-of-all-trades Multi-domain Wildcard SSL certificate, which functions as a Multi-domain SSL certificate as well as a Wildcard certificate. Depending on the issuing CA, you can secure between 25 and 250 domains on a single certificate. This certificate will also cover all the subdomains of the domains you are securing with this certificate. In simple terms, this certificate is all you need to secure your entire web presence.

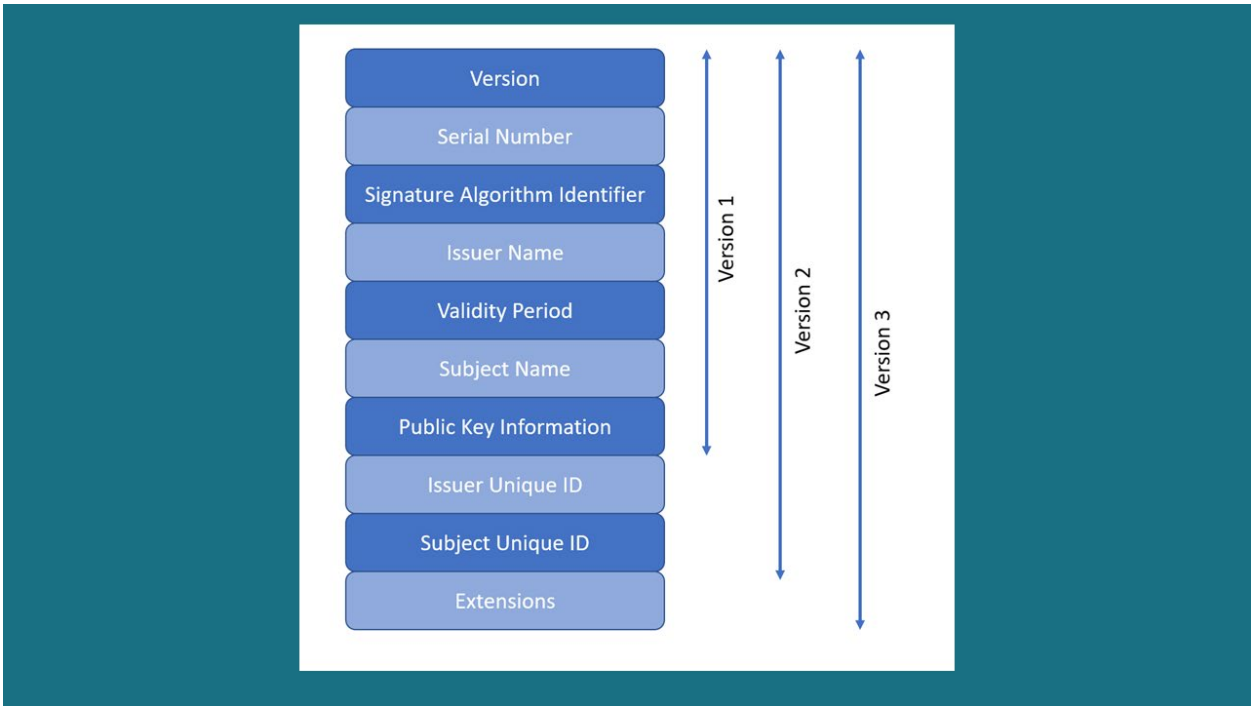
Following are domains and subdomains a multi-domain wildcard SSL certificate can secure.

- ✓ My-New-Domain-1.com
- ✓ www.My-New-Domain-1.com
- ✓ My-New-Domain-2.org
- ✓ My-New-Domain-2.net
- ✓ blog.My-New-Domain-2.net
- ✓ My-New-Domain-3.com
- ✓ My-New-Domain-1.edu

Here, you will get two options in one single SSL Certificate. Other than protecting multiple domains, you will also be able to secure subdomain levels.

SSL/TLS: Structure & Formats

An SSL Certificate is also known as X.509 certificate, where X.509 is a set of standards which defines the structure of the certificate. It helps in defining the data fields which need to be included in the SSL/TLS Certificate. It uses a formal language known as Abstract Syntax Notation One (ASN.1) to express the data structure of the certificate.



X.509 v3 Digital Certificate Structure

Since the establishment of the X.509 certificate in 1998, three different versions of the X.509 public-key certificate standards have been evolved.

Version 1 of X.509

X.509 Version 1 certificate contains the below mentioned basic fields:

Version: Specifying the number of encoded certificates, where currently the values of this field are 0,1 or 2.
Serial Number: It contains a unique and positive integer assigned to the certificate by the CA (Certificate Authority).

Signature Algorithm: It contains an object identifier (OID) specifying which algorithm will be used by the CA to sign the certificate. For example, 1.2.840.113549.1.1.5 specifies an SHA-1 hashing algorithm in combination with the RSA encryption algorithm.

Issuer: It contains the X.500 DN (Distinguished Name) of the certificate authority that created and signed the certificate, where X.500 is a computer network standards' series that covers the services of an electronic directory.

Validity: As the name implies, contains the validity period, i.e., issuance date and expiry date of the certificate.

Subject: Contains the name of the entity related to the public key contained in the certificate as per the X.500 distinguished name.

Public Key: Contains information related to the public key and its associated algorithm.

Version 2 of X.509

X.509 version 2 certificate contains the below mentioned fields along with the basic fields of Version 1.

Issuer Unique Identifier : It's an optional field which contains a value to uniquely identify the certificate authority.

Subject Unique Identifier : It's an optional field which is used to provide a unique ID for the subject.

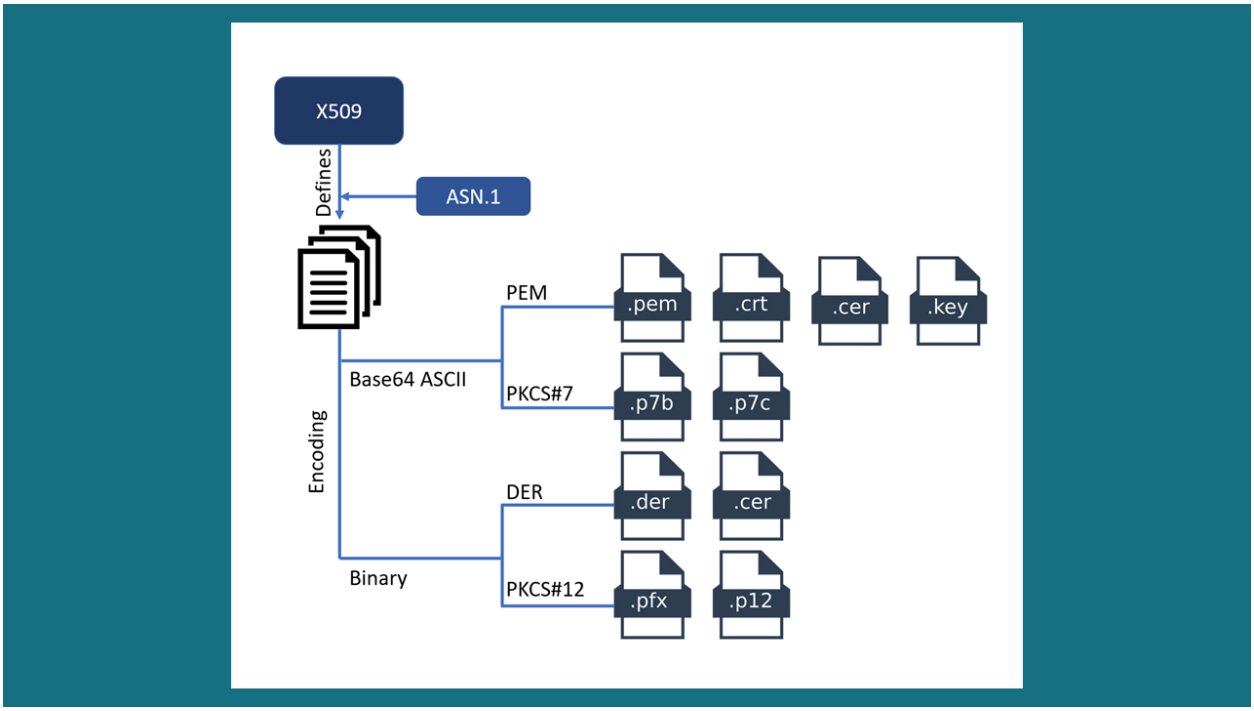
Version 3 of X.509

Along with Version 2 and Version 1 of X.509 certificate, it contains an additional field extension which is used to support different applications.

Extensions : It is defined as a series of data type Extensions, where every extension has three different parts, namely Extension ID (extnId), Critical and Extension Value (extnValue).

X.509 Certificate File Format

Furthermore, there are different X.509 certificate formats like DER, PEM, PKCS#7 and PKCS#12. CAs will provide the certificates with one of these formats. Here, PKCS#7 and PEM formats use Base64 ASCII encoding & DER and PKCS#12 use binary encoding. Likewise, all the certificates have different extensions based on their used encoding and format.



SSL/TLS Certificate Format

PEM Format

Usually, CAs (Certificate Authorities), provide certificates in PEM format which are encoded files in Base64 ASCII. The file type of this certificate can be .crt, .pem, .cer or .key. And this .pem file can include the server certificate, the intermediate certificate and the private key file within a single file. It's also possible that the server and the intermediate certificate can be provided in a separate file, .crt or .cer and the private key in a .key file.

PEM files can be opened through text editors like notepad and MS word, as it uses an ASCII encoding. Also, the PEM file contains the certificate between the statements ---- BEGIN CERTIFICATE---- and ----END CERTIFICATE----. The private key is between the ---- BEGIN RSA PRIVATE KEY----- and -----END RSA PRIVATE KEY----- statements and the CSR is between the statements -----BEGIN CERTIFICATE REQUEST----- and -----END CERTIFICATE REQUEST-----.

PKCS#7 Format

The PKCS#7 format is a Cryptographic Message Syntax Standard which uses a Base64 ASCII encoding file with .p7b or .p7c extension. Also, only this certificate can be stored and not its private keys. This certificate is contained within the statement -----BEGIN PKCS7----- and -----END PKCS7-----.

DER Format

DER Certificates are mainly used for Java-based web servers and they are in binary form with an extension of .der or .cer files.

PKCS#12 Format

The PKCS#12 certificates are mostly used in the Windows platform and they offer two different extensions of files, .pfx and .p12. It uses a binary form and helps to store the server certificate, the intermediate certificate and the private key within a single .pfx file with password protection.

All SSL/TLS Certificates come with a limited period and once it gets over, they will no longer be considered valid. Certificates have different validity periods, which could be one or two years. They are set to expire by that date. Certificates must be replaced once they expire to avoid security warnings or disruptions. And this process is known as Renewal of SSL Certificate.

Granted, the usage of SSL/TLS Certificate is widespread, and it's used throughout the organization, but there are many reasons to consider a *Lifecycle Management Approach* as it's one of the critical tasks to maintain an accuracy of SSL/TLS Certificates which should not be taken lightly. Let's understand in detail.

What is an SSL Certificate Lifecycle Management?

As said, SSL Certificates comes with limited validity and don't allow updates like any software. But SSL management throughout the network ensures protection and prevents failures, some of the basic needs of the businesses. By employing a proper *SSL/TLS Certificate Lifecycle Management*, it ensures a proper approach and increases the effectiveness as well as efficiency. Likewise, the SSL certificate requires proper management, which is divided into certain stages as below:

- ✓ **Enrolment of Certificate**
  Enrolment of SSL Certificate is done as per the request the user makes to the trusted CA. It's a cooperative process between the CA and a user. The enrolment request contains enrolment and the pubic key information. Once the user requests the certificate, the CA verifies all the information. Based on its policy rules, the CA creates and posts the certificate and sends a certificate to the user. During this distribution, the CA sets certain policies which affect the user of the certificate.

- ✓ **Validation of Certificate**
  Whenever an SSL Certificate is used, the status of that certificate is verified to know whether that certificate is valid. During this process, the CA checks and verifies the status of the certificate to be sure that it's not in its Certificate Revocation List (CRL).

- ✓ **Revocation of Certificate**
  All the certificates issued by the CA comes with a validity period. In case the certificate needs to be revoked before its expiry date, the CA can be instructed to add the certificate into its CRL. If the person responsible for handling the SSL leaves the company, if the certificate is lost or if the private key or the certificate gets compromised, the certificate could be revoked.

- ✓ **Renewal of Certificates**
  Once the certificate reaches its expiration date, it must be renewed automatically or manually by users. Also, while renewing a certificate, you must choose whether to generate new private and public keys.

- ✓ **Abolishing Certificates**
  If the certificate is no longer needed, then that certificate and any backups related to that certificate must be abolished with its associated private key. In return, it helps in ensuring that the certificate is not used or compromised by anyone.

- ✓ **Inspecting Certificate**
  Tracking the creation, revocation, and expiration of certificates come under certificate inspection. Sometimes, it also tracks every successful usage of a certificate.

If the organization fails to maintain proper lifecycle management of the certificate, it can face negative consequences such as expiry of the certificate, loss of reputation and revenue.

Some other important steps that involve in the SSL/TLS Certificate lifecycle are:

- ✓ CSRs
- ✓ Validation Process
- ✓ Expiration
- ✓ Renewal

CERTIFICATE SIGNING REQUEST GENERATION

Many assume that the whole process is complete after they purchase the SSL certificate. But, that's not the case. Purchasing an SSL/TLS certificate is just the first step. The main process begins after that, which is the generation of CSR (Certificate Signing Request).

What is A Certificate Signing Request?

A Certificate Signing Request (CSR) is one of the request blocks in an encoded text format which is provided to a Certificate Authority when a Client applies for an SSL/TLS Certificate. Usually, it's generated on the same server where the SSL/TLS Certificate has to be installed, and it contains information which will also be included in the SSL Certificate such as the name of the Organization, Common Name (Domain Name), Locality and Country.,

Also, CSR plays a crucial role in the issuance of an SSL/TLS Certificate, as it's used by a CA (Certificate Authority) to create your SSL Certificate. But as mentioned above, it does not have a private key. Additionally, SSL Certificate created with a particular CSR only works with its matching Private Key, which was generated along with it. So, if you lose your Private Key, the

Certificate will no longer be considered secure, and it's recommended that you create a new one by replacing or reissuing your SSL/TLS Certificate.

Information Contained in Your Generated CSR

Name	Description	Example
Common Name	It's an FQDN (Fully Qualified Domain Name) of your server that must match exactly with the information you provided in your web browser or else you will receive a name mismatch error.	*.domain.com mail.domain.com
Organization	The legal/official name of your organization without suffixing or using abbreviations like Corp, Inc or LLC.	ABC Example, Inc.
Organization Unit	The division or department of your organization that handles the certificate.	IT Department Information Technology
City/Locality	The name of the city where the organization is located. Do not use abbreviations.	Spokane
State/Country/Region	The state or region where your organization is located. Do not use abbreviations.	Washington
Country	The two-letter ISO code of the country where your organization is located.	US
Email Address	An email address that is used for contacting your organization.	webmaster@google.com
Public Key	The public key which goes along with the certificate.	It is created automatically

What Does a CSR Look Like?

Generally, a CSR is created in the PEM format encoded in Base-64 that can be opened using a text editor such as Notepad. However, it's a must to include the header and footer lines "-----BEGIN CERTIFICATE REQUEST-----" and "-----END CERTIFICATE REQUEST-----" at the beginning and the end of the CSR.

Sample of a CSR

```
-----BEGIN CERTIFICATE REQUEST-----
MIICtjCCAZ4CAQAwcTELMAkGA1UEBhMCVVMxEDAOBgNVBAMMB2FiYy5jb20xFDAS
BgNVBAcMC1NhbiBBbnRpb25vMRwwGgYDVQQKDBNBQkMgTmV0d29yayBQVlQgTFRE
MQ0wCwYDVQQIDARUZXhhMQ0wCwYDVQQLDAROZXdzMIIBIjANBgkqhkiG9w0BAQEF
AAOCAQ8AMIIBCgKCAQEAuEnyMukefVsyxCjfo200FJBR92nn4sOXPTrspncw8ji8
F6PI+Ld6FqwQ5cgqaKzMC02qNSqLy4yqyD4eSubT24ECpbEM/ko6ZBDOZQOloI5n
v5tWAiCrJlMcH5Aso2qbD9m8ZCG1IPqPQ6+uuAfpPKNqEUxWV0LNtfYTd2VdpmIm
jH26B0QiVEVBpjJbj7uF9IeVmwUfIbIzde8WycVa+8IrqAyUlDP5YMgwqwCZQm3/
mjmu7irZc12VpnUaZVWBbvXbsJwVcpBQq8pQkJe5yEDLq1j7ctYBUlCiYg6T/Ant
/9hqifh8olnykaKrUur+WQYyE8EK+5iuquymkOC4eQIDAQABoAAwDQYJKoZIhvcN
AQELBQADggEBAIEr3ehVV6rlJuOQY0IQC3oHw55b3WWRSNvaQJtMTGtHqOEuPTNJ
JkVALR+FR0Hk2HACPsEBUjSXRRj5ibtph0DlP3Khi/0G7yWrSw/vLRWrnkunAmtU
nIq+Buyem+ssaprYwsaaay7N16u2ZmEiHboU2APWWm5RuD31Csn1INGIlPs3InH0
fkN9Kb1xU2sIEnwXqA7Y+ouZ38iBuuI3Pyc+icT1iii9pZI/tGQHp3RDaL3+4OQ4
U8GrsruPAA4+qSdOqMVuaINf6IirJTBy15xgLdHzviniSSCH2IMNEze3ahLnshox
NXASSDIxBOclS9Av1Dv/TiGAKQsNJMRZIQM=
-----END CERTIFICATE REQUEST-----
```

Furthermore, it is advisable to *refer to CSR Generation guides*, as instructions differ from one server to another.

SSL VALIDATION PROCESS

The validation process is one of the crucial aspects of the SSL/TLS Certificate. Whenever someone purchases an SSL Certificate from the Certificate Authority, the CA requires the website or the organization to go through a mandatory vetting process. The screening process gets rigorous based on the type of the SSL certificate purchased.

Moreover, there are three different validation levels and all have a different process. Let's see each one in detail.

DV SSL: Validation Process

Domain Validated (DV) SSL/TLS Certificate is one of the most basic SSL Certificates and the validation steps are also simple compared to Organization Validated (OV) and Extended Validated (EV) SSL Certificates. To get a Domain Validated SSL/TLS Certificate, you just need to confirm that you own the domain for which you are purchasing an SSL Certificate. Following are the options to complete Domain Validation.

- ✓ **Email Based Verification:** It's one of the easiest and preferred verification methods. The Certificate Authority (CA) will send an email to the WHOIS registered email

address asking you to verify that the certificate is registered for the specified domain name. Once the email verification is done, the validation will be complete, and the certificate will be issued in minutes.

The authentication email might be sent to one of the five pre-approved email addresses that are generally associated with the website.

- Admin@name-of-site.com
- Administrator@name-of-site.com
- Webmaster@name-of-site.com
- Hostmaster@name-of-site.com
- Postmaster@name-of-site.com

✓ **File-Based Verification:** If you choose this method, the Certificate Authority (CA) will provide you with an HTML file called Authentication File (auth file), which includes the Hashed content. You simply need to upload that file onto your server directory and verify it. Once it's verified by the Certificate Authority, your SSL Certificate will be issued.

✓ **CNAME-Based Authentication:** This method is used only for SSL/TLS Certificates purchased from Comodo. Comodo will provide you with two unique hashes, one using the MD5 algorithm and another using the SHA2 algorithm. Once you receive these values, you have to enter them as your CNAME DNS record. For that, the format is: <MD5 hash>.yourdomain.com CNAME <SHA-2 hash>.comodoca.com. Once you complete this step, Comodo will verify it.

OV SSL: Validation Process

As the name implies, the Certificate Authority (CA) will verify whether the organization for which the OV SSL/TLS Certificate is purchased is a legal entity.
Every business must fulfill the following requirements:

✓ **Organization Authentication**
The Certificate Authority (CA) checks whether the organization is registered and active within the state or country. Additionally, all the information you provided about your organization must match the organization's registration details. It's also important to note that all the registration information like the organization operated under any trade names, assumed names or DBAs must be up to date and accurate, as well.

✓ **Locality Presence**
To fulfill this requirement, the CA (Certificate Authority) will verify that the Organization has a legit physical presence within the country or in the state it's registered. To verify all this information, the Certificate Authority will look through the Online Government

Database or any other relevant database and check all registration details such as city/state/country.

- ✓ **Telephone Verification**
  The organization must have an active telephone listing that can be verified by looking at an acceptable online telephone directory.

- ✓ **Domain Verification**
  Here, the domain name will be verified, and the organization has to prove that it owns the website. To verify, the CA will look through the WHOIS registry and the internet database that stores the information of the registrar. Apart from this, for this process to work, the record must be available publicly and it should also display the verified business name with the corporate identifier with a physical address, as well.

- ✓ **Final Verification Call**
  It's one of the simplest requirements for an Organization Validated SSL Certificate. To complete this requirement, the Certificate Authority will call the phone number associated with the organization and ask certain easy-to-answer questions such as "Did you order this?" or "What is the name of your company?," to verify the order details. Moreover, if the given phone number does not connect directly to your desk, there are other ways such as IVR Extension or Transfer or Alternate Number, that are accepted during the Final Verification Call process.

For legal entities, it shouldn't be a problem if they fail to meet the requirements as per the expectations as there are other methods, as well.

Official Registration Documents

You can provide official business registration documents issued by the government. For example,

- ✓ Articles of incorporation
- ✓ Chartered licenses
- ✓ DBA statements

or any other government document which shows your company is a legal entity.

Dun & Bradstreet

It's an organization well-known for the financial reporting of businesses. CAs hold their credit reports in high value and they are used to meet multiple requirements like

- ✓ Operational Existence
- ✓ Physical Address
- ✓ Telephone Verification

Their reports are also used to verify specific details associated with the business entity.

Legal Opinion Letter

Legal Opinion Letters, also known as Professional Opinion Letters (POLs), are quite hard to obtain as an attorney or an accountant has to vouch for the legitimacy of your organization by signing a letter. These letters are very useful when it comes to the validation process as they satisfy multiple requirements.

- ✓ Organization Authentication
- ✓ Operational Existence
- ✓ Physical Address
- ✓ Telephone Verification
- ✓ Domain Authentication

Third Party Directory

CA's can use an existing or a new telephone listing of an accepted third-party directory, such as

- ✓ Yellow Pages
- ✓ Scoot
- ✓ 192.com

The business name and physical address in the listing and that on the enrollment form and your company's certificate must be the same.

Note: The Certificate Authority can choose any document from the given list to fulfill the validation requirement and it can also differ from one to another.

EV SSL: Validation Process

An Extended Validated (EV) SSL/TLS Certificate offers the highest level of authentication and security compared to the Domain Validated and Organization Validated SSL/TLS Certificates. It goes one step further to offer more trust by verifying the existence of business to let users know that the website is genuine and trustworthy.

It provides the highest possible security, its validation process is also rigorous compared to the other two, i.e., Domain Validated SSL Certificates and Organization Validated SSL Certificates. Moreover, no matter which Certificate Authority you choose to purchase from, the validation requirement of an EV SSL/TLS Certificate is almost the same because of the CA/B (Certificate Authority and Browser) forum which is a regulatory body run by the CAs (Certificate Authorities).

Validation Requirement for EV SSL Certificate

- ✓ **Enrollment Form:**
  The enrollment form, also known as Acknowledgement of Agreement, is the very first and easiest requirements a business or an organization has to meet to get an Extended Validated SSL/TLS Certificate. You simply need to fill the form that requires basic information such as:

  - The Organization's name
  - The Organizational contact's official title
  - The full name of the Organizational contact
  - The signature of the Organizational contact
  - Date & place of signing
  - A contact in HR to verify that the Person (Organizational Contact) who has applied for the certificate is employed within the company.

  The main purpose of this form is to verify that the Organizational Contact has the right to act on behalf of the mentioned Organization, as no one would like to give out an EV SSL Certificate to an imposter website.

- ✓ **Organization Authentication**
  The Certificate Authority verifies that your company is a legal entity which is active and registered in the local municipality.

- ✓ **Operational Existence**
  Here, the Certificate Authority will confirm that your company has been operating for three or more years. If its existence is more than three years, it is a breeze as there's a chance that you won't have to provide any additional documents and it will be covered with Organizational Authentication via online checking.

- ✓ **Physical Address**
  The Certificate Authority will ask you to prove that your Organization is well established, and it has a physical presence within the country or state where it's registered. To complete this requirement, the Certificate Authority will verify the complete address of your company through the Online Government Database that includes the street address, city, state, and country.

- ✓ **Telephone Verification**
  The Certificate Authority will ask for an active telephone number. First, it will look through the Online Government Database and if it matches with the other information, this step will be completed without any difficulty.

- ✓ **Domain Authentication**
  It's a straightforward approach where the CA will confirm that your company legally owns the domain for which the EV SSL/TLS Certificate was ordered. The very first way is to check through the details available in WHO.IS. (a website which displays the domain registrar information). If the record of your company is publicly available and it shows the exact information such as the verified business name and physical address as mentioned in the Enrollment Form, you won't have any trouble.

- ✓ **Final Verification Call**
  This requirement is easy as it takes place after the completion of the prior requirements. The Certificate Authority calls you on the verified business telephone number and asks simple questions such as "Did you order this?" or "What is the name of your company?" to verify the order details.

  The CA will try its best to verify each and every requirement. But if the information is not found, or if it's not updated in the Online Government Database, there are few other ways to fulfill those requirements.

- ✓ **Email or Fax the Documents**
  If due to any reason, the Enrollment Form requirement is not completed, you can send the paper version of the form through mail to the CA's physical address, or it can even be faxed.

- ✓ **Official Registration Documents**
  If you fail to meet the Organization Authentication, Operational Existence or the Physical Address requirements, you can provide documents issued by your local government as proof. For example
    a. Articles of Incorporation
    b. DBA Statements
    c. A Chartered License

- ✓ **Dun & Bradstreet**
  Credit reports from well-known and reliable financial reporting companies such as Dun & Bradstreet are accepted by CAs to fulfill the following requirements.
    a. Operational Existence
    b. Physical Address

c. Employment Verification
d. Telephone Verification

✓ **Legal Opinion Letter**
A Legal Opinion Letter or a Professional Opinion Letter (POL), is a letter given by an accountant or attorney that has vouched for the authenticity of an organization. It helps fulfill the following requirements.
a. Organization Authentication
b. Operational Existence
c. Physical Address
d. Telephone Verification
e. Domain Authentication

✓ **Recognized Third-Party Directories**
It's an alternative for the telephone verification requirement. Here, Certificate Authorities will use the telephone listings in a recognized third-party directory like
a. Yellow Pages
b. Scoot
c. 192.com

✓ **Bank Confirmation Letter**
In case the establishment of your organization is less than three years old, the Certificate Authority would accept a letter from the bank telling your organization has an active checking account (Demand Deposit) with them.

Note: Certificate Authority can choose any document from the given list to fulfill the validation requirement.

SSL CERTIFICATE EXPIRATION

Why Do SSL Certificates Expire?

Several times, people raise questions regarding the shorter validity of an SSL certificate and relate them to some scam of making money. It's even valid to ask such questions as the annual bills are something hard to ignore. But again, the reason for the expiration of SSL/TLS certificates is to keep websites updated with all the latest security standards. This way, you will always have the latest TLS ciphers and versions.

Moreover, technology advancement happens very often, and SSL certificate is not an exception. By having an expiry date for SSL certificates, users are enforced to stay updated with all the latest developments, with updates happening periodically.

If the certificate does not have a validity period, it will become hard to keep it updated with all the latest encryption technologies, which makes it open to cyber-attacks due to loopholes in outdated encryption standards.

For example, there are two latest updates, SHA-2 and the latest TLS1.3. Everyone has the privilege to enjoy the benefits of these two updates, and if the certificate does not have an expiry date or has a longer duration like a decade, it will become next to impossible to keep the customers updated till the certificate expires and they will be open to cyber-attacks.

What happens when your SSL certificate expires and how to Avoid it?

Many times, people forget to do things on time even if it's important. But there are some tasks that should not be delayed or else it can cause a negative impact. Renewing a website's SSL/TLS Certificate is one of those. Primarily, if your website deals with financial transactions such as banking or online shopping, it becomes more vital to avoid this situation.

Among many services, the security service provided via SSL/TLS Certificate also comes with a specific expiry date, which shouldn't be missed. Every SSL/TLS Certificate has a validity period of 1-2 years. It could be one year or two years. Once that validity period is over, the certificate needs to be renewed and if you fail to do so, it will expire.

Moreover, if a website owner allows the SSL/TLS Certificate to expire, it will result in an invalid certificate and the website will no longer be considered safe to do transactions. So, you can imagine how important it is to avoid the expiry of an SSL/TLS Certificate and how it can become a problem if someone fails to do so.

But again, sometimes, especially in big enterprise-level businesses, this type of scenario is possible. Mainly, due to oversight or maybe because the contact person is moved, being promoted or maybe some other reason. Because, when it comes to enterprise businesses, they do not deal with one or two, but several SSL/TLS Certificates, which is not that easy to maintain.

Apart from this, Website users also face problems while accessing websites with expired certificates such as warning or error messages that will eventually result in the loss of trust and also decline in sales & revenue. This will ultimately have an adverse effect on the brand and reputation of the business.

So, how to prevent an SSL from getting expired? Below are some of the actionable tips that may be helpful to you:

- ✓ Find a good certificate management platform. Visibility is one of the biggest challenges faced by enterprise businesses. It's a fact that website owners may not remember the expiration dates of their SSL certificates. To cope with these types of

situations, Certificate Authorities like Comodo and DigiCert offer a platform that helps enterprises to manage SSL Certificates and the entire infrastructure.

- ✓ Take your time and decide which CA(s) you would like to work with and then set up CAA records for restricting who can issue SSL for your domains. This will help you eliminate the chances of new rogue certificates from getting issued.
- ✓ Track email addresses used for purchasing SSL/TLS Certificates. Set reminders.

One thing to note is that the expiry of SSL certificates plays an essential role in offering trust to customers. Many times, organizations forget to renew their SSL certificates on time, which can create uncertainty in the minds of website visitors. It's recommended renewing your SSL before it gets expired rather than losing your valuable customers.

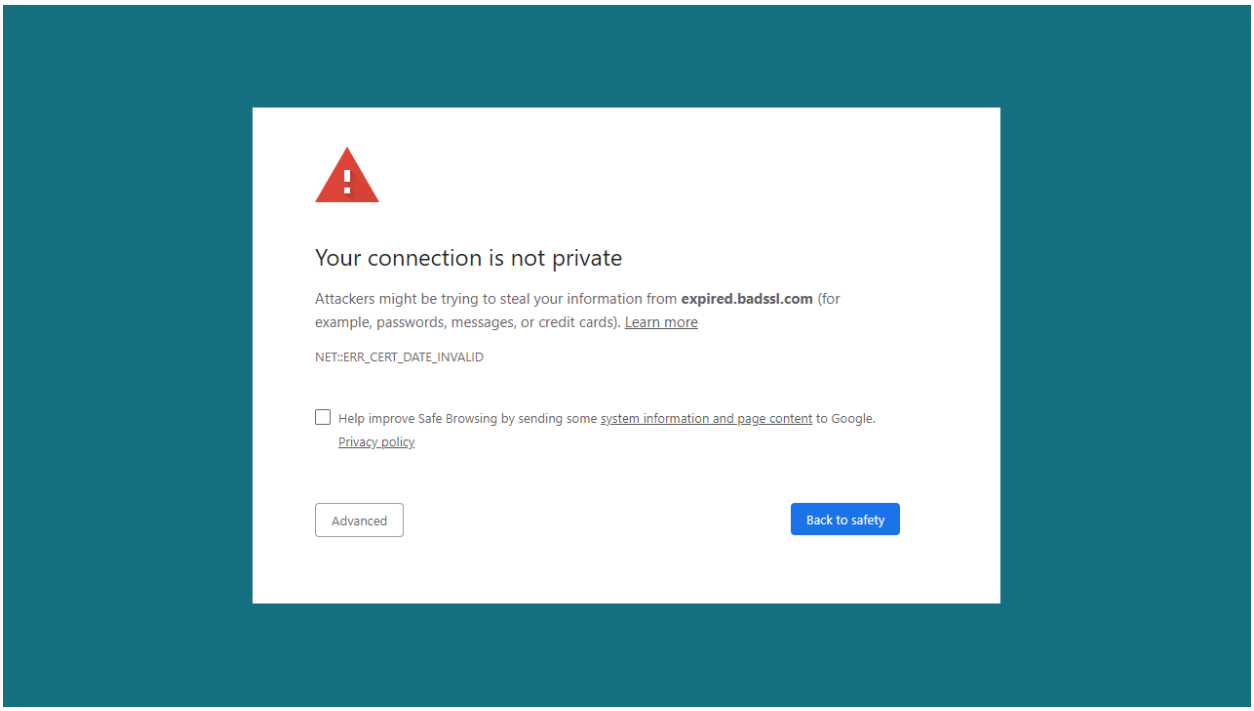
HOW TO RENEW SSL CERTIFICATE

Renewal of SSL Certificate is one of the crucial parts of the certificate lifecycle. Like passports, SSL certificates also come with an expiration date and any attempt to use these SSL Certificates for authentication before or after the validation period will fail.

SSL/TLS Certificate renewal is the process in which a user buys a new certificate for the same public key which was used in the expiring SSL certificate. Mostly, the SSL/TLS Certificate expires after one or two years from the purchase date. To maintain the trust of site visitors, renewing a certificate within the last quarter of the certificate's validity period is essential. Likewise, by renewing your SSL/TLS Certificate, you also reap the benefit of keeping the latest and updated SSL/TLS versions and ciphers.

Moreover, the renewal process of SSL/TLS Certificate is simple and if you're an existing customer of any certificate provider, then there's no need to worry about the expiration date. As, it's the responsibility of the system to remind you about the expiration date, to give a smooth experience while avoiding the last-minute hassles. SSL Certificates can be renewed before 90 days of expiration, reminders also start coming during that period.

Also, the reasons for renewing your SSL Certificate before its expiration plays an important role. For instance, no one would like to lapse the protection of their website. And, if you go for renewing your SSL Certificate once it expires, you're not renewing it but purchasing it all over again. CA's will make you go through the entire process you went through when you initially purchased your SSL certificate. Likewise, until the process is complete, your website will remain unprotected.



Warning Message When SSL Certificate Expires

On the other hand, if it's renewed before the expiration of a certificate, the workload of Certificate Authority (CA) will also lessen down. Most of the information from the previously validated certificate will be rolled over by the CAs, depending on the certificate type if the business information has not been altered. And sometimes, you also get the benefit of not dealing with the entire process, but to go through a few important ones.

Lastly, you should remember that once you renew the certificate, your job will not be over until you install it. Newly issued certificates still have to be validated by the CA or else it will keep you unprotected when the installed certificate expires.

SSL REVOCATION

Certificate Authorities (CAs) allow revocation of SSL/TLS certificates. It is also one of the crucial and ill-understood parts of enterprise security. If a CA revokes an SSL certificate before its expiration period, it'll no longer be trusted. To be more precise, revocation of SSL Certificate means to invalidate an already issued certificate. Also, these revocations of the certificates are presented through attestations signed by the CAs that issued the certificates. Likewise, they are also responsible for publicizing the status of the revocation for all the certificates issued by them.

Besides, whenever an SSL connection is established by a client, as a part of the SSL handshake, the server provides the chain of certificates. In addition to that chain verification, the correct behaviour of the client also includes confirmation of all the certificates belonging to that chain

are not revoked before that SSL connection continues. And, all the certificates come with data pre-included regarding where and how to check for the revocation information.

The Need of SSL/TLS Certificate Revocation & How to Address it?

Besides checking the validity period of the certificate, it's also essential to check whether the certificate has been revoked or not. There are numerous reasons which make the revocation of certificate necessary. For example,

- ✓ If a person responsible for handling certificates leaves the company or changes his role
- ✓ If an improper certificate is issued by the CA
- ✓ If the domain is no longer owned by its owner for whom it was issued
- ✓ If the original certificate is replaced with a different one from another issuer
- ✓ Owner of the Certificate stops operations or if the site is no longer active

Moreover, different methods are available for querying and publishing these lists. But many are not used widely, due to the slow and prone to fail methods or because it's not easy to understand and implement.

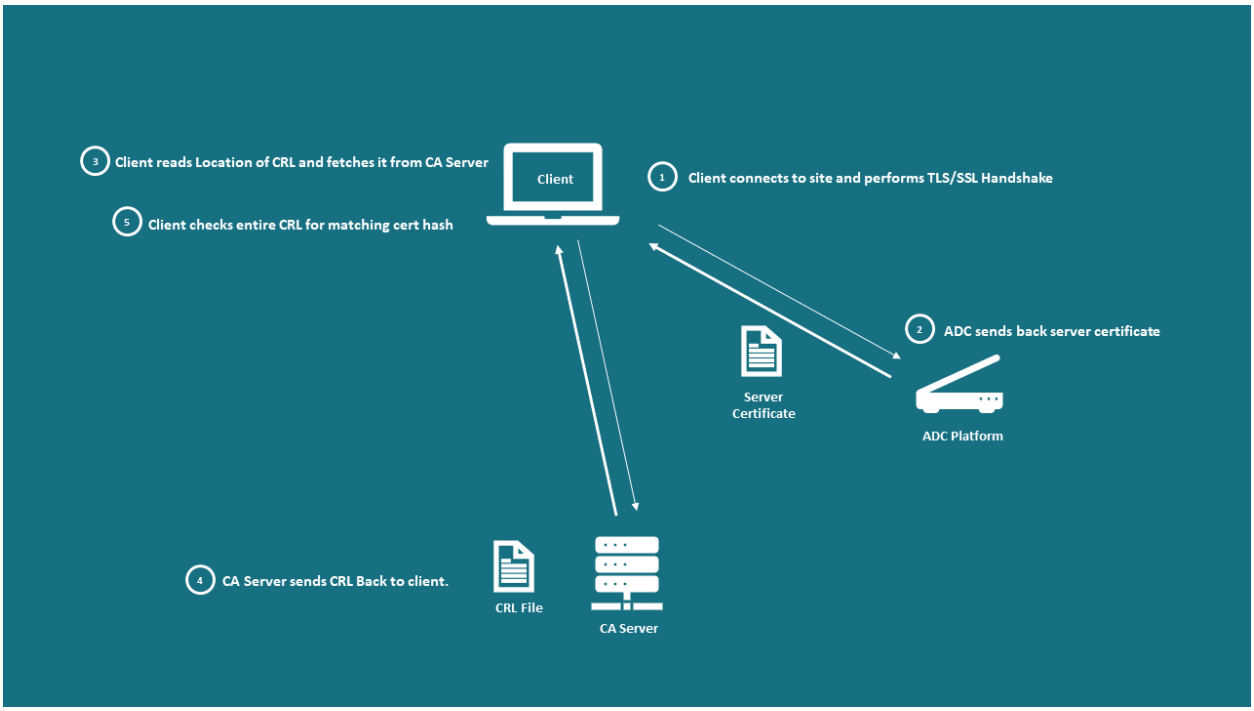
CRL (Certificate Revocation Lists)

CRL is one of the basic forms of revocation checking methods., Here, a text file is created by the CA (Certificate Authority), which contains a list of (serial number, revocation timestamp, revocation reason) data record/certificates before their expiration, which is signed by the CA. Also, an authenticating device like a web server or ADC (Application Delivery Controller) checks this text file containing a list for every session where it must be authenticated. Also, it's acceptable for the user to proceed further if the presented certificate is valid and does not appear on the list.

Likewise, CRLs do share some critics as well. They are simple to use but challenging to implement, operationally. Moreover, they are not even updated regularly, making it difficult to manually import in authenticating devices like ADC. Lastly, they can grow in large numbers, making it difficult to consider because of the multiple CRL sources.

CRLDPs (CRL Distribution Points)

CRL Distribution Points are configured to overcome the CRL problem of manually importing a CRL file so the webserver or ADC can automatically read it from an online source, usually LDAP or HTTP(S). Though, one of the larger problems of CRL, which is size, still exists.

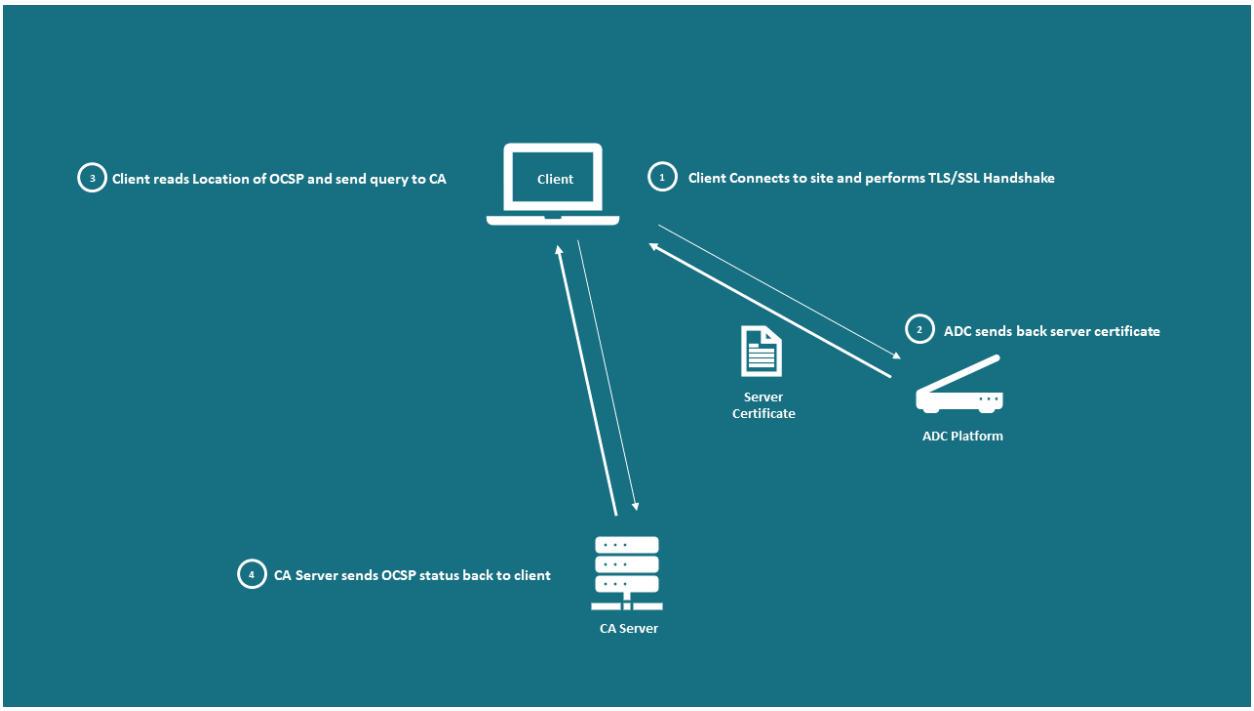


Process of CRLDP

For instance, a 2015 paper [An End-to-End Measurement of Certification Revocation in the Web's PKI] presented that, though the new CRL files might only be in a large number of bytes, the average CRL file for Certificate Authorities can be anywhere from 0.5 MB to 7MB. And, in an eCommerce environment, every user would also go through a manual process of downloading the list for getting ensured that the site they are getting connected with does not have their certificate revoked.

OCSP (Online Certificate Status Protocol)

OCSP is an improvement and solution to the problem of size, faced in CRL. Here, the process is quite similar to CRL checking, but the difference is that the client only has to fetch the status of the specific certificate compared to the entire list.



Process of OCSP

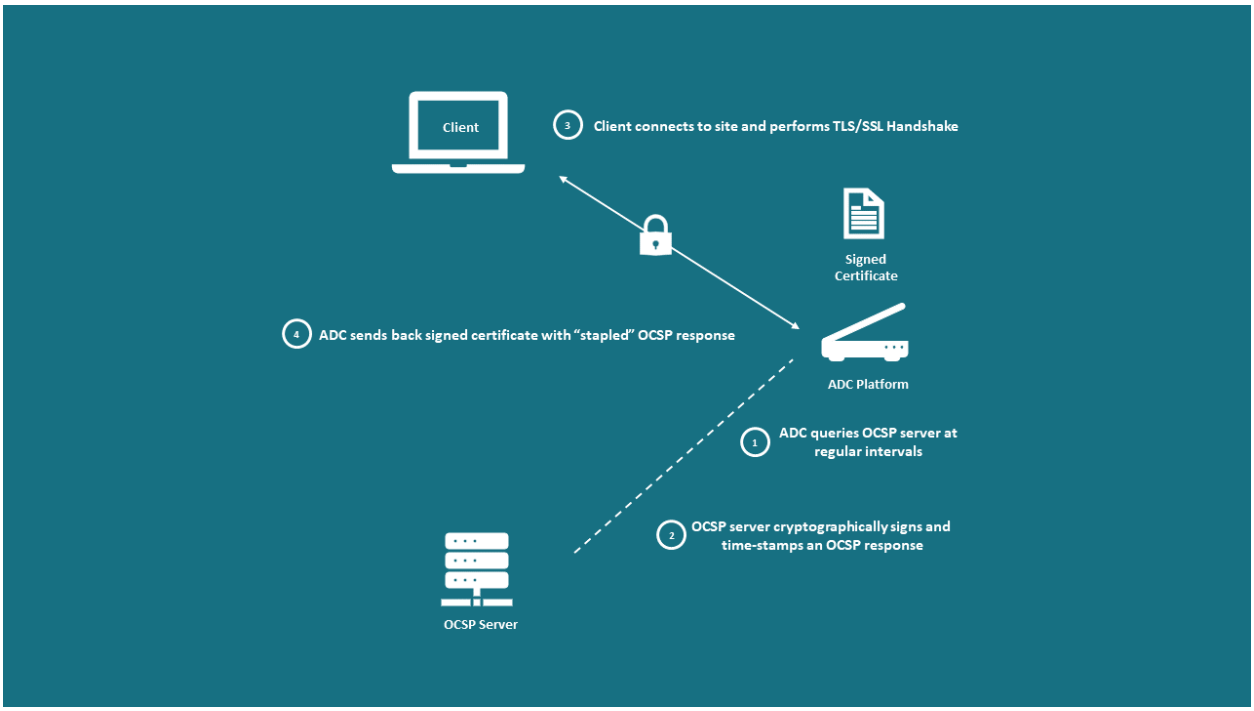
Though the OSCP checking process is more efficient for clients to check the certificate's revocation status, still it consists of certain issues such as:

- ✓ As every client has to check the revocation status of every certificate, the number of queries hitting the CA server (OCSP responders) will get high.
- ✓ As the OCSP responder has the log of IP addresses and the websites visited by every client, it can create a privacy leak.
- ✓ The entire process already gets slow because the client has to go through another series of round trips for connecting and querying the status of the certificate. If there's a poor mobile network or any other high latency connection, then the delay will be furthered more.

Additionally, the main issue of the revocation methods is that the client handles all the burden. Every user and web browsing request must have their query for the revocation service. OCSP stapling is another approach to this.

OCSP Stapling

In OCSP Stapling, the burden of proof is moved from client to the ADC or web server. For example, ADC periodically fetches the OSCP status for the certificate it manages. And the OCSP responder result is digitally signed, so if there's any tampering, it may get spotted.



Process of OCSP Stapling

Here, the client has to make a single request to the web service. As the ADC sends back the digitally signed status of the certificate, like the certificate for the website it's delivering is sent back. Moreover, OCSP response is short-lived, and they are also approved by the CA, making them trustworthy to clients.

Some of the things improved by OCSP Stapling are:

- ✓ It removes the amount of time spent on page load by eliminating the requests of the OCSP service
- ✓ Fixes the privacy issue, as now the OCSP responder gets requests only from one place, ADC or the webserver
- ✓ Also cut downs the overhead on the OCSP responder, as it now only serves infrequent requests of the webserver or ADC

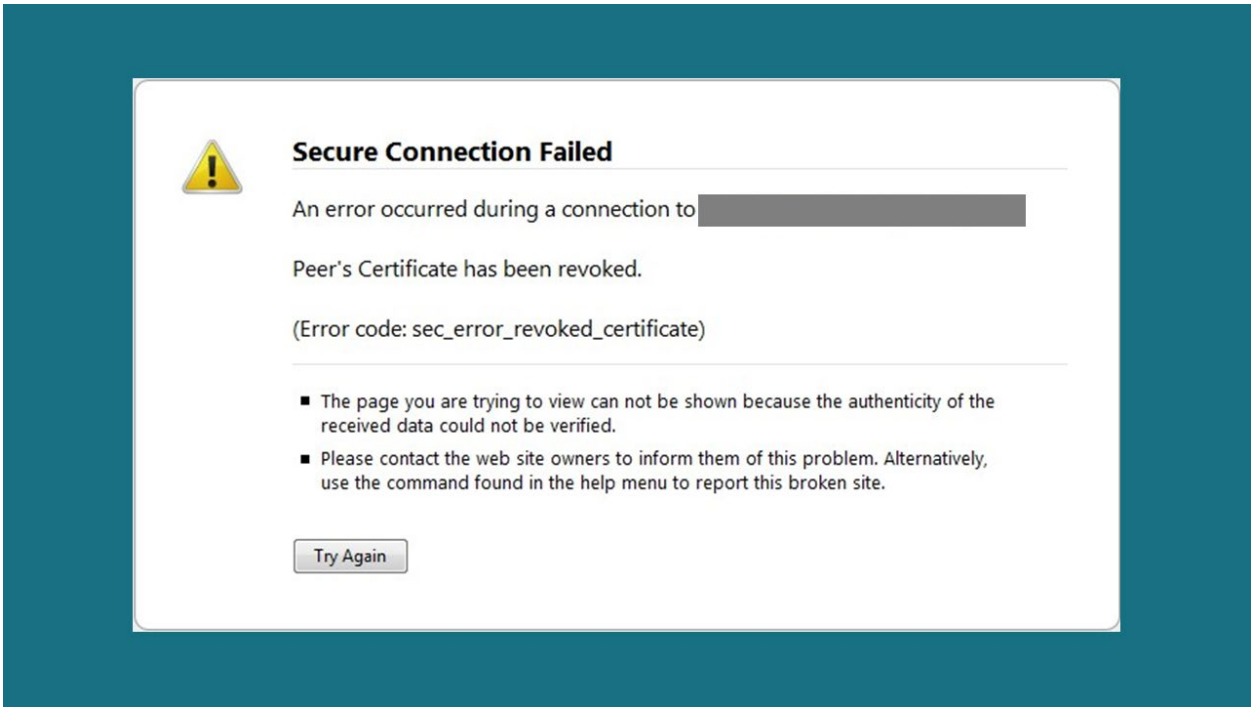
Still, OCSP Stapling also has some issues like chances of active man-in-the-middle (MITM) attacks like in OCSP. Also, OCSP stapling is not mandatory but an option, as it's possible to block the OCSP response and trick the web browser into thinking that the malicious certificate is genuine.

Certificate Revocation in Browsers

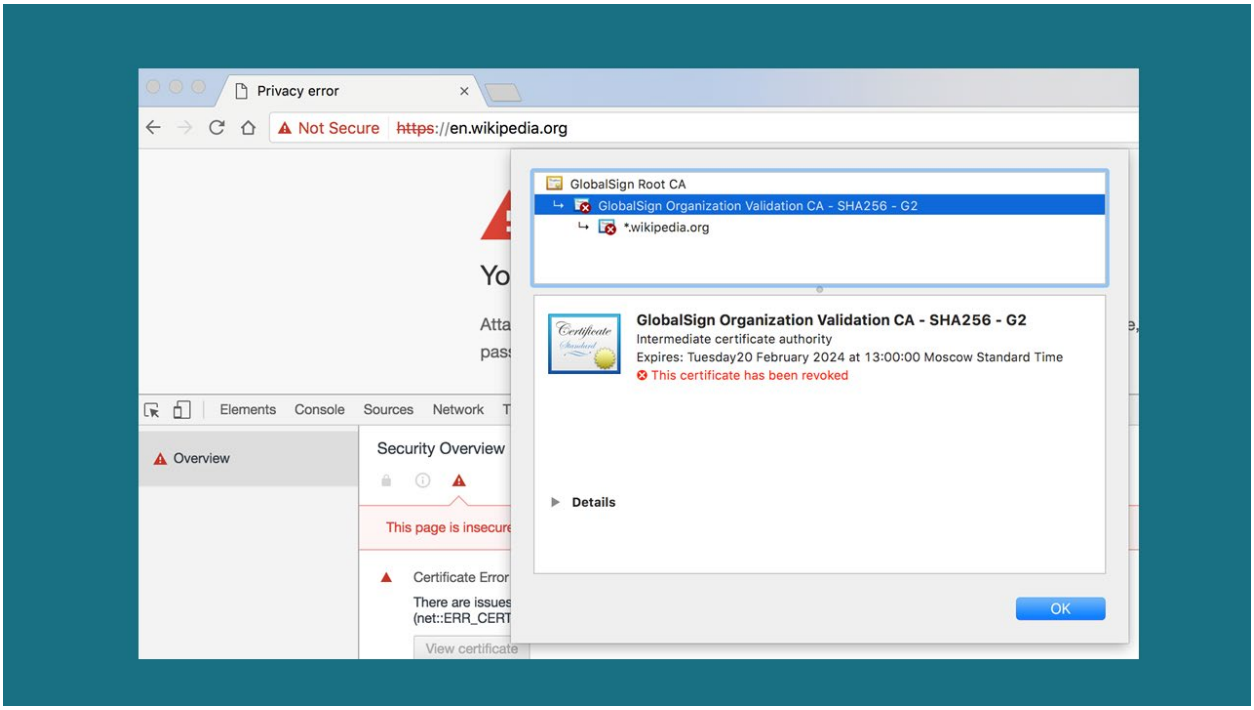
Popular web browsers like Google Chrome and Mozilla Firefox have their approach towards certificate revocations.

For example, Google Chrome does not use CRL lists or OCSP servers, but they have their CRLset, which is the list of revoked certificates compiled and embedded inside the Chrome browser. They are auto updated by periodically crawling the CRLs of all the major CAs around the globe. And Mozilla Firefox has an analogue solution known as OneCRL and also uses the regular OCSP approach.

Firefox usually shows SEC_ERROR_REVOKED_CERTIFICATE error like below, when the revoked status is encountered from OCSP responder:



And as OSCP is not used by Google Chrome, it has to be looked deeper to know whether it has been revoked, as shown below:

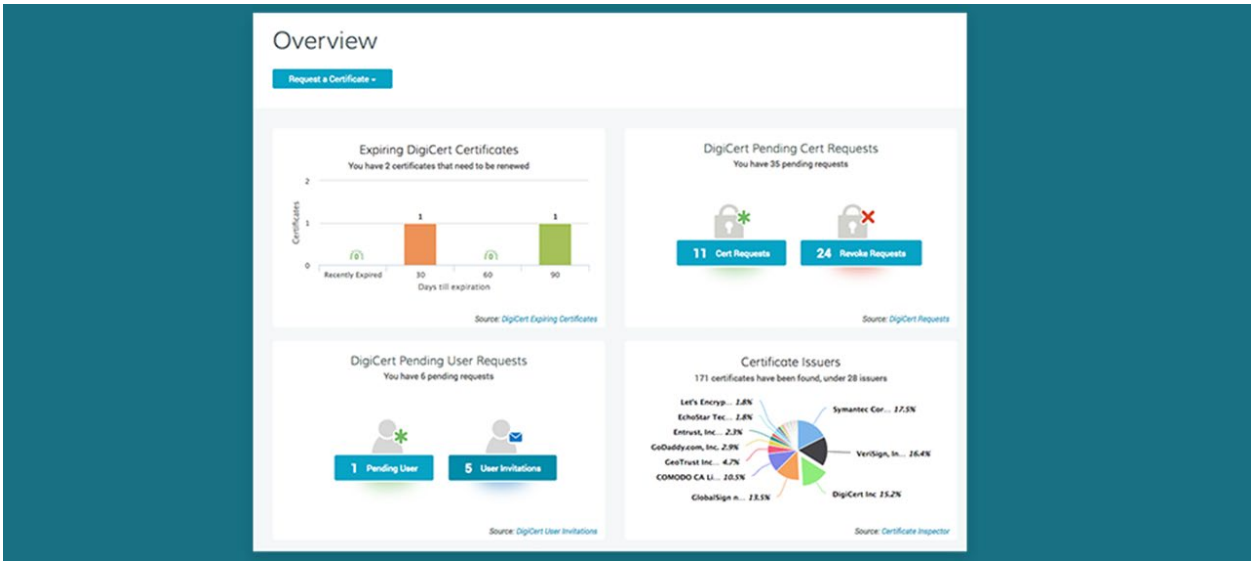


Again, there is not a 100% reliable solution for browsers to detect revoked certificates on time. But, when it comes to end-server certificates, it's suggested to follow security best practices, such as by purchasing reduced validity period certificates. So, the certificates can be renewed frequently, while reducing the timeframe for an attacker to make use of the stolen certificate. Also, it is advised to keep private keys safe by not allowing the CAs to generate your private key and to keep it password protected.

Lastly, revocations and expiration of the certificates are entirely different., Expired certificates are considered invalid and not every active certificate is not valid either. Revocation and other validation techniques are an essential part of all the proper PKI procedures, as in real-world environment, there's a chance of making mistakes in vetting and in the certificate management process.

Managing one or two SSL Certificates may not be an issue. But, in large organizations, managing a large number of SSL Certificates can become an arduous task if it has to be done manually.

Managing SSL certificates is not that time consuming as everyone thinks. With the right guidance and tools like a centralized platform which can help in automating the certificate management operations and offer timely insights on an environment of organization's SSL, managing the SSL Certificate lifecycle becomes quite easy.



Many Certificate Authorities like Sectigo and DigiCert provide a complete SSL management platform which helps you to manage the SSL Certificate at scale. Also, it helps in automating the discovery, issuance, renewal, replacement and revocation of multiple certificates.

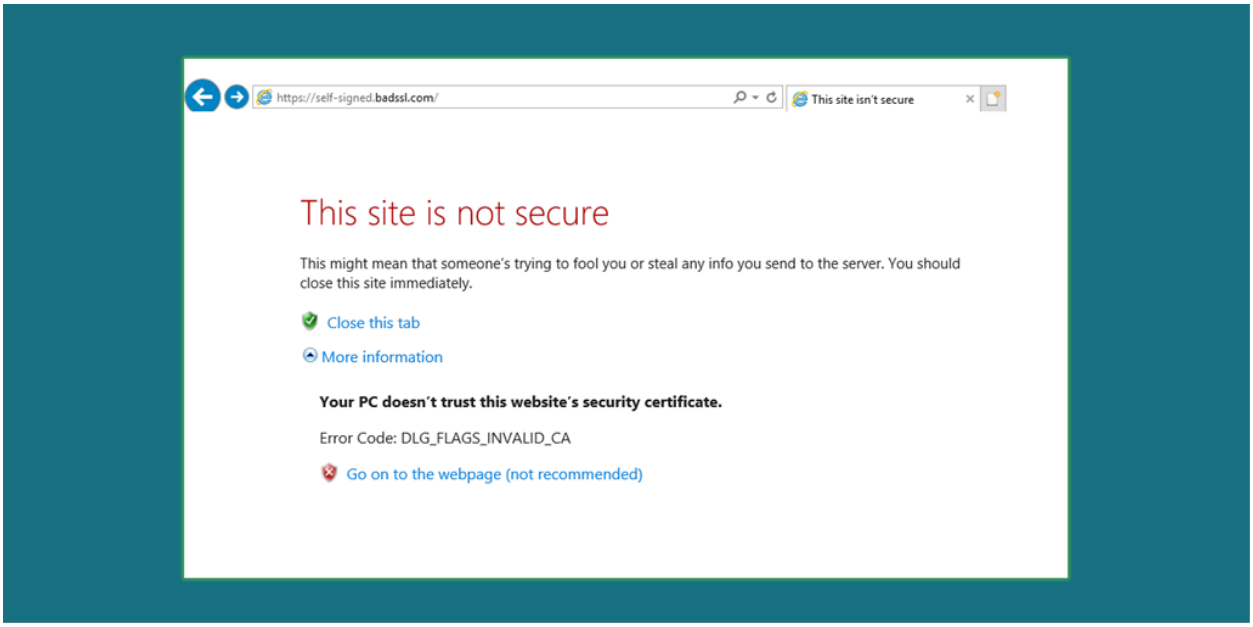
Features of SSL Management Tools:

- ✓ A single management interface to improve speeds while simplifying the issuance, deployment, discovery, and renewal of all certificates, no matter where it's located in your organization.
- ✓ Ability to communicate with all the leading operating systems, devices, protocols, and chipsets while allowing you to secure your infrastructure, no matter which technology is used.
- ✓ Automated management feature which helps in performing discovery and installation, renewal and replacement.
- ✓ Reduced staff time and operational costs associated with individual certificate management

APPENDIX A: COMMON SSL/TLS MISTAKES

Looking to install a certificate onto your web server? If you're not a tech-savvy person who is aware of SSL Certificates, then it's fair enough to say that it will be quite a hassle for you if you try to install it on your own. Again, here are some of the common mistakes that you should avoid while *installing an SSL/TLS Certificate*.

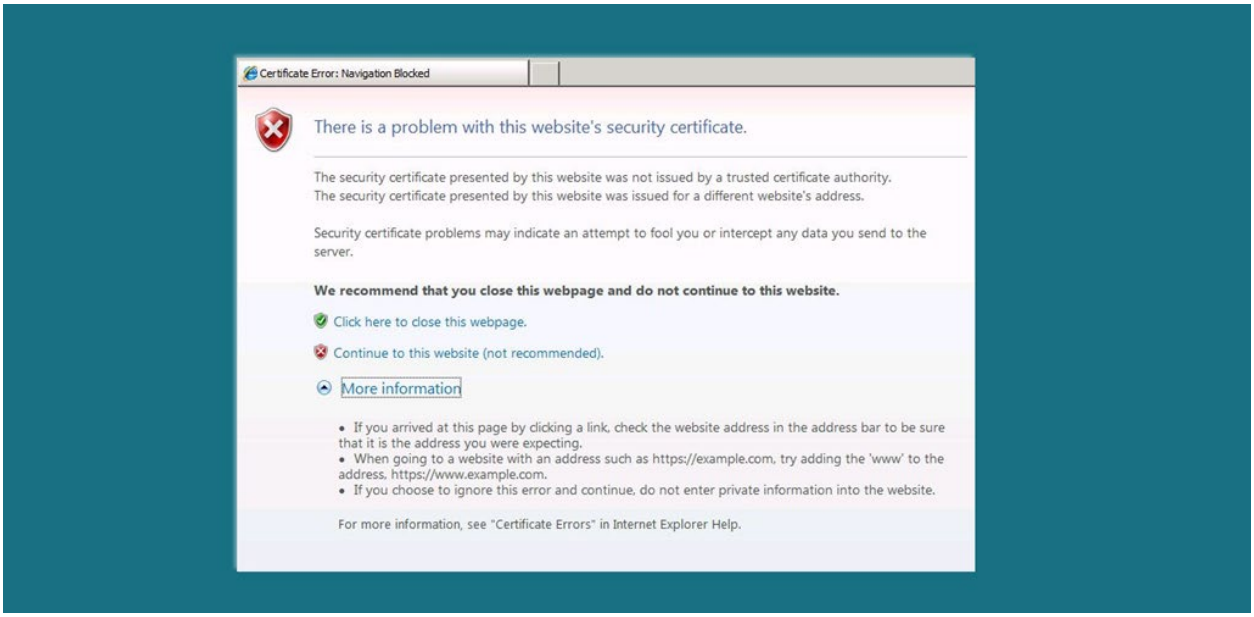
Using Self-Signed SSL Certificates



Short and simple, avoid these certificates unless you're using it for testing purposes within an organization. A self-signed SSL Certificate will be authenticated by you and not by a certificate authority. In other words, you will be trying to make others believe you, "just because you said so" without any legitimate proof, which is futile as no one will believe it.

In the same way, web-browsers will not accept it and will also warn your website visitors by displaying a warning message saying, "The site's security certificate is not trusted!", eventually preventing users from visiting your website.

Choosing an Untrustworthy Certificate Authority



Some Certificate Authorities may not be trusted by web browsers. If you have chosen to go with such Certificate Authorities, you will inevitably face errors quite similar to that of the Self-Signed SSL/TLS Certificate Errors, where web-browsers warn visitors whenever they try to visit your site. It's better to avoid SSL Certificates from such CAs and it's even true to a certain extent that all Certificate Authorities are not equal. The best way to deal with such situations is to go with branded Certificate Authorities as their security solutions and other trust indicators like site seals are more recognized rather than going for something else just for the sake of saving money.

Mistake in CSR (Certificate Signing Request)

Generating a Certificate Signing Request (CSR) is one of the crucial steps. While installing an SSL certificate, it is mandatory to complete this step without a single mistake. The CSR generation process differs depending on the server you are using, so it's better to thoroughly go through everything to generate the CSR. You will not be able to install your SSL/TLS certificate if you make mistakes or enter incorrect information. Apart from this, it's also recommended that you verify the CSR using tools such as CSR Decoder once you finish it, as it will be helpful to know whether you should move to another step or not.

Not Fully Prepared for the Validation Process

Validation is one of the mandatory steps. Before the SSL/TLS Certificate gets issued to you, the Certificate Authority verifies you and your organization. If it's a Domain Validated SSL/TLS Certificate, it will only verify the WHOIS registry information and you should respond through the email registered with it. For the Organization Validated (OV) and Extended Validated (EV) SSL/TLS Certificates, the process is a bit complex, as you have to provide certain proofs for verification.

So, if anything goes wrong, for instance, if your organization does not have a telephone number which is publicly listed, then there could be a delay in the issuance of your SSL certificate and in the worst case, it may not get issued at all.

Problems with Your Private Key

After completing the CSR creation process, an additional file named 'Private Key' will come along with it. This private key is an essential element that helps in unblocking the encrypted communication transit between your visitor's web browser and your webserver. Losing or sharing your private key might land your website in trouble, and you will have to contact your CA to get it reissued. So, better safe than sorry!

Not Following the Installation Guide Properly

If you're not an IT professional, it's better to go through guides that can help you install an SSL/TLS Certificate. Whether you purchase your certificate from a reseller or a Certificate Authority, they do offer knowledge base which consists of several guides and tutorials on SSL/TLS certificate installation for different web servers.

Once You Encounter a Mistake, You Don't Contact Customer Support Service

You might already be aware of a certain mistake you made but you might try to solve it on your own. It's good to take matters into your hands, but if you're one of those inexperienced persons for whom SSL is quite new, rather than taking risk, it would be better to contact Customer Support Service. Certificate Authorities and SSL Certificate Providers offer chat, email and telephone support. They even offer installation services at a lower price. So, why not get done

with it within a short period rather than spending endless hours on errors about which you're not aware of?

You Didn't Check Once the Installation Is Over

Once the certificate is installed you might want to test it to make sure that your time has not been wasted. It's also recommended that you check whether the SSL certificate is installed properly, because at times you may assume that it has been installed properly and your site is secured and trustworthy for your website visitors, but in reality, it may not be so. It's better you do a basic test by visiting your website through an HTTPS protocol, for example, by typing "https://domain-name.com" in the address bar to verify whether the padlock appears just to confirm it has been properly done. You can also use certain tools like SSL Checker to get more specific results.

You Forgot to Renew Your Certificate

The SSL/TLS Certificate you purchase and install comes with a specific validity period of 1 or 2 years. Once it's over, they expire. The main reason to keep an expiry date is to keep the customers updated with the latest evolving security standards by continually authenticating their identity.

In other words, it's mandatory that you keep renewing the certificate to enjoy the benefits of a secured website. But sometimes, it happens and even big companies like LinkedIn, Yahoo and Google forgot to renew their SSL Certificates, which made the websites temporarily unsecured. Popular web-browsers such as Mozilla Firefox may even block such unsecured websites. To avoid such situations, it's better to remember SSL renewal dates and renew it before it gets expired. SSL providers also send email reminders, so don't take it lightly.

APPENDIX B: COMMON CERTIFICATE ERRORS

Before we get into different types of SSL Errors, let's see what exactly each error means. In simple words, an SSL Error is a problem faced by users due to SSL/TLS Certificates such as getting warning messages when they try to access a website.

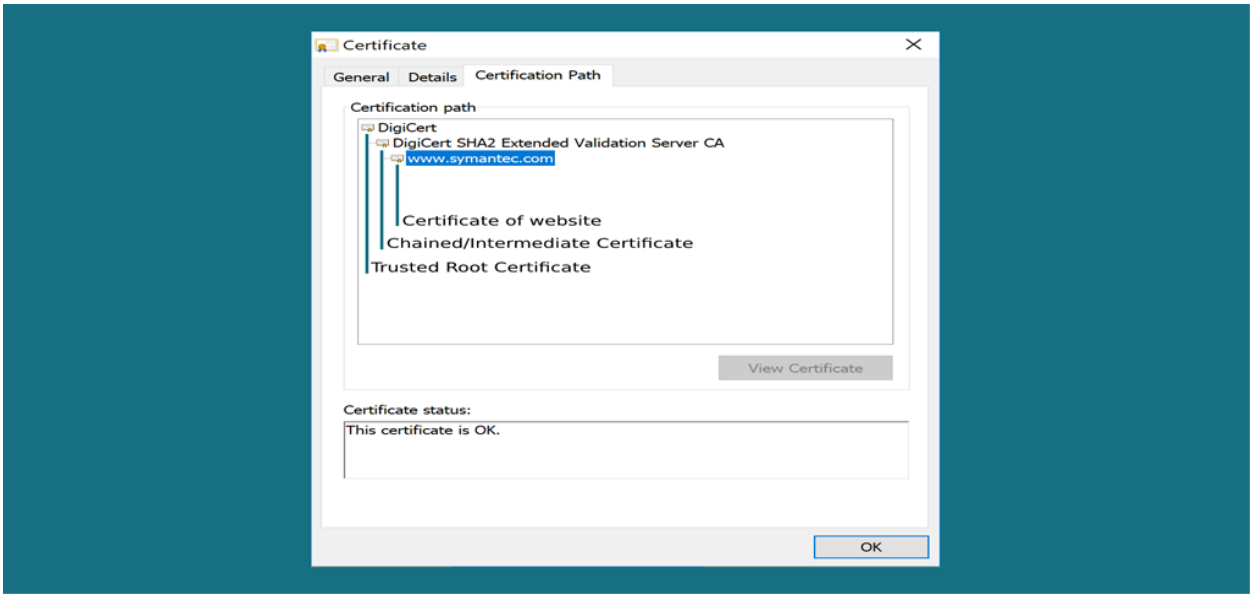
SSL is one of the protocols used to encrypt information sent to a web server from a web browser. So, when something prevents your computer from initiating this secure session with the website you want to access, an error is displayed, which is known as an SSL Error.

There are more than one SSL Errors, let's see them one by one:

This SSL certificate is untrusted

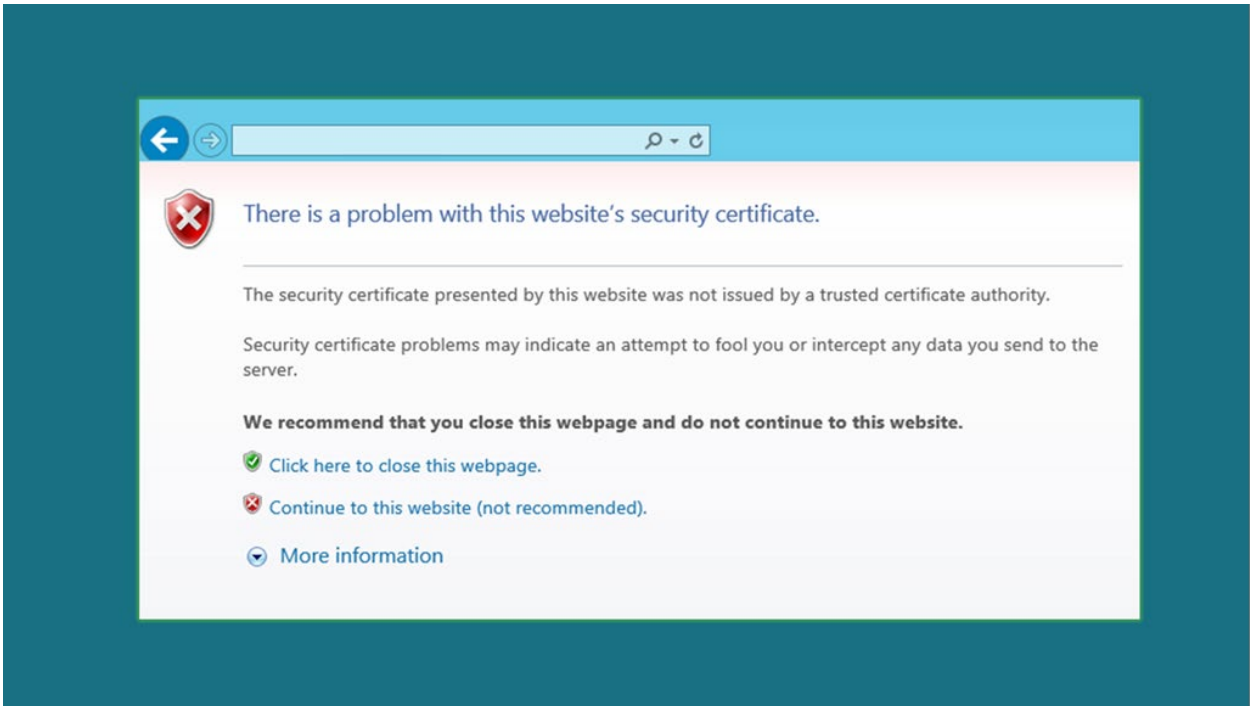
If you see the warning message "This site's security certificate is not trusted," it generally means that the browser has failed to link the certificate of that website with the trusted root certificate or it's not signed by a trusted root certificate.

If a trusted certificate authority (CA) has signed the certificate, then the chain/intermediate certificate may not have been installed on the web server in between the primary and the root certificates.



To verify the "Certification Path," open "Certificate Details" in internet explorer and click on the "Certificate Path" tab.

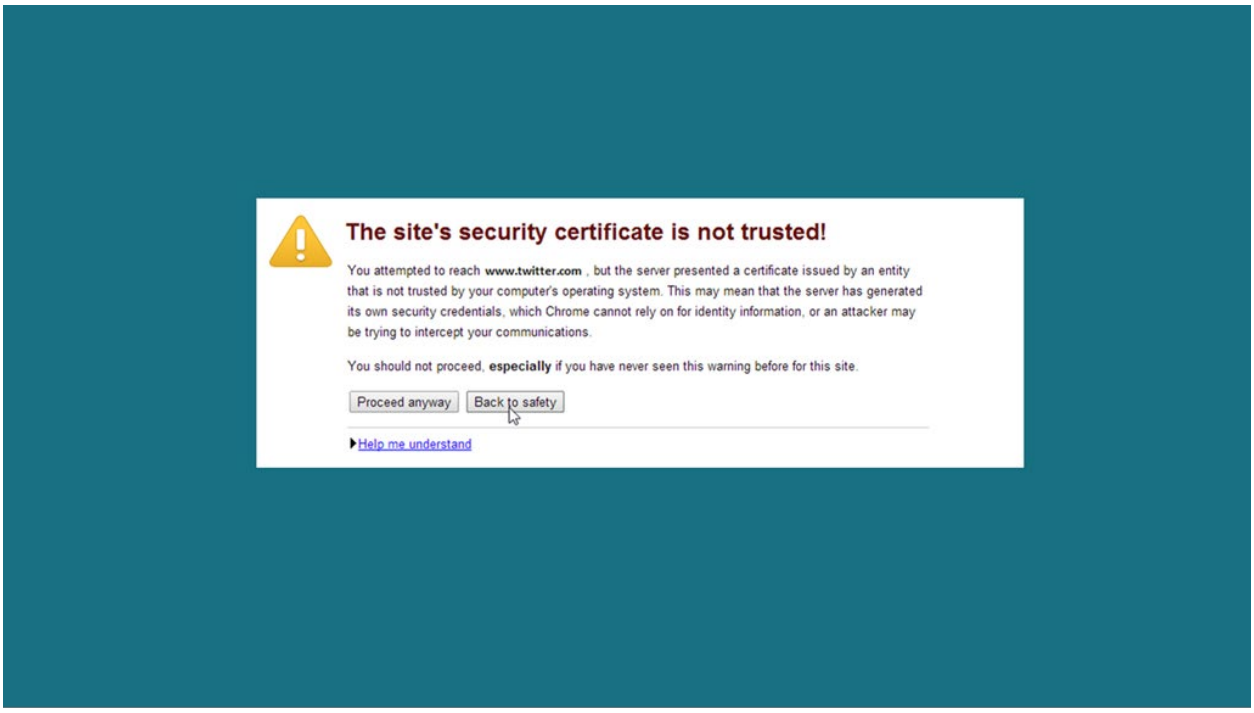
If you face any problem while installing the chain/intermediate certificate, the best bet is to contact your Certificate Authority.



The security certificate was not issued by a trusted certificate authority

If you see this error, it means the SSL/TLS Certificate is not approved or signed by a trusted company, according to the web browser. This could be due to the following reasons:

- ✓ **Self-Signed Certificate Used by a website:** Though a self-signed certificate is created free of cost, it's not as trusted as an SSL Certificate issued by a trusted third-party.
- ✓ **Free SSL/TLS Certificate is installed on the website:** Free SSL/TLS Certificates are offered by certain Certificate Authorities that are trusted by all major web browsers, for example, Google Chrome & Mozilla Firefox, but for a free SSL/TLS Certificate, the Root Certificate has to be imported manually to overcome this error.
- ✓ **Intermediate/Chain Certificate is missing, though the trusted SSL is installed on the website:** Most probably, all the trusted certificates will ask you to install minimum one chain/intermediate certificate on the server to link your certificate to a trusted source.



The Site's Security Certificate has expired

All SSL/TLS Certificates come with a validity period of 1-2 years. If the certificate is not renewed before it expires, it will expire and it will cease working, which will lead to this error. The simple solution is to renew it.

The connection between the browser and the website might not be secure

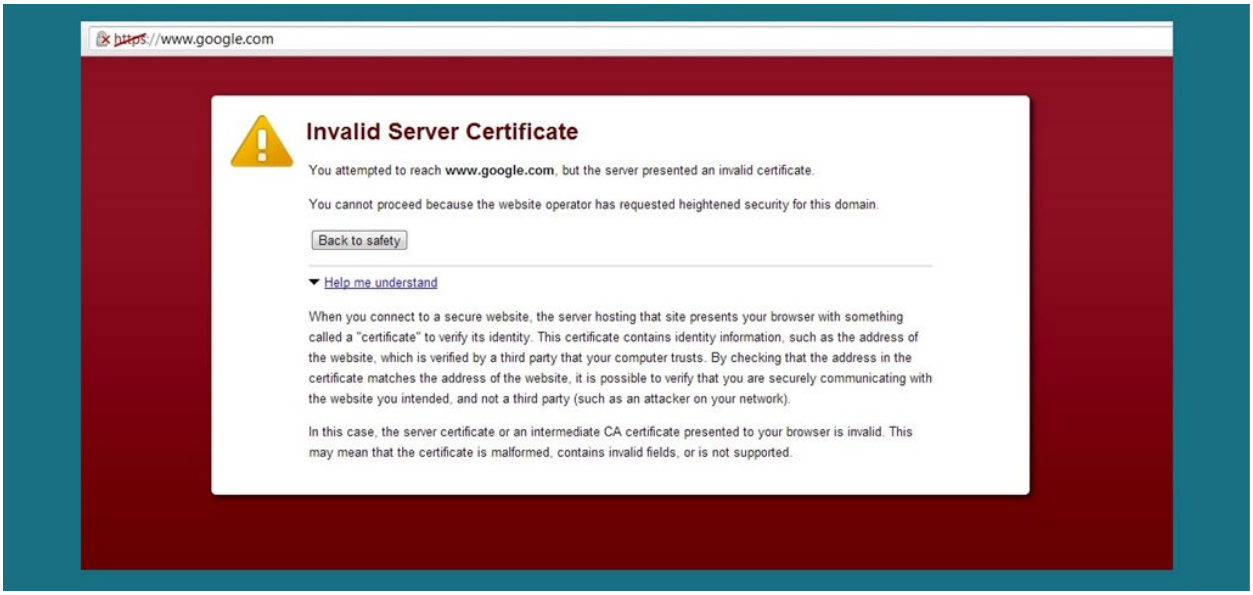
You might be trying to access a website on an unsecured public Wi-Fi connection. This may lead to the display of this error on your web browser.

This page contains both secure and nonsecure items

Generally, this error occurs when certain elements on a secured page (page loaded with https:// in the address bar) are not loaded from a secure source. Maybe because, images, frames, javascript, and iframes are loaded from HTTP and not HTTPS. To be more specific about the same, you can *use a tool like WhyNoPadLock,* which will help you to locate the problem.

Certain steps you can take to solve this issue are:

1. Change all URLs to https
2. Change all links to // or make them relative
3. Change the settings of your web browser



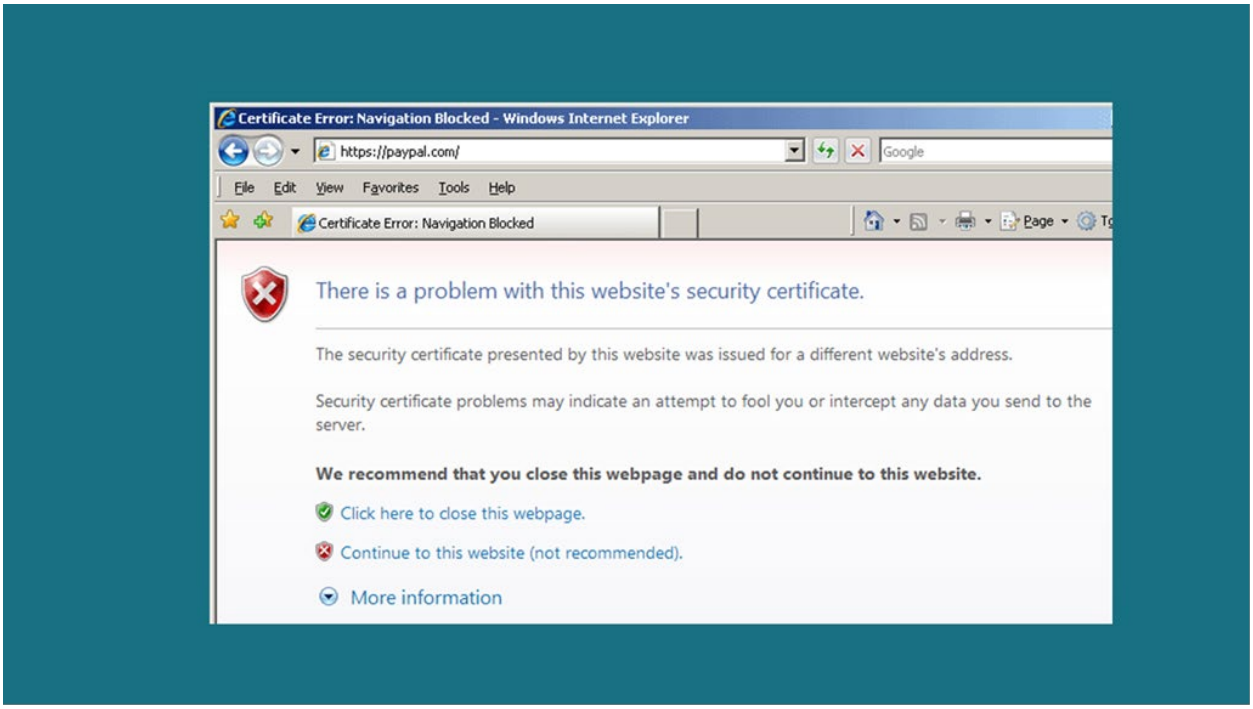
Invalid Server Certificate Error

It's one of the errors which is displayed when the web browser fails to find a valid server certificate to start a secured communication, as a server certificate issued by a trusted Certificate Authority is mandatory to start an encrypted session.

To solve this error, the following operations may be helpful:

- ✓ Clear Web Browsing History and Cookies.
- ✓ Check whether the system time zone matches with the current time zone. Reset, if it doesn't.
- ✓ Check whether the SSL/TLS Certificate is installed properly using the SSL Checker tool. If it's not installed properly, install it again.
- ✓ Check Firewall and Antivirus definition.i If the website is blocked, unblock it.

‘SSL Certificate Name Mismatch’ error



SSL Certificate Name Mismatch Error generally occurs when the listed common name on an SSL fails to match the name displayed in the URL bar. To establish a successful encrypted connection, the names mentioned on the SSL Certificate and the name in the URL should be same.

More to add, different web browsers show the same error but with different messages.

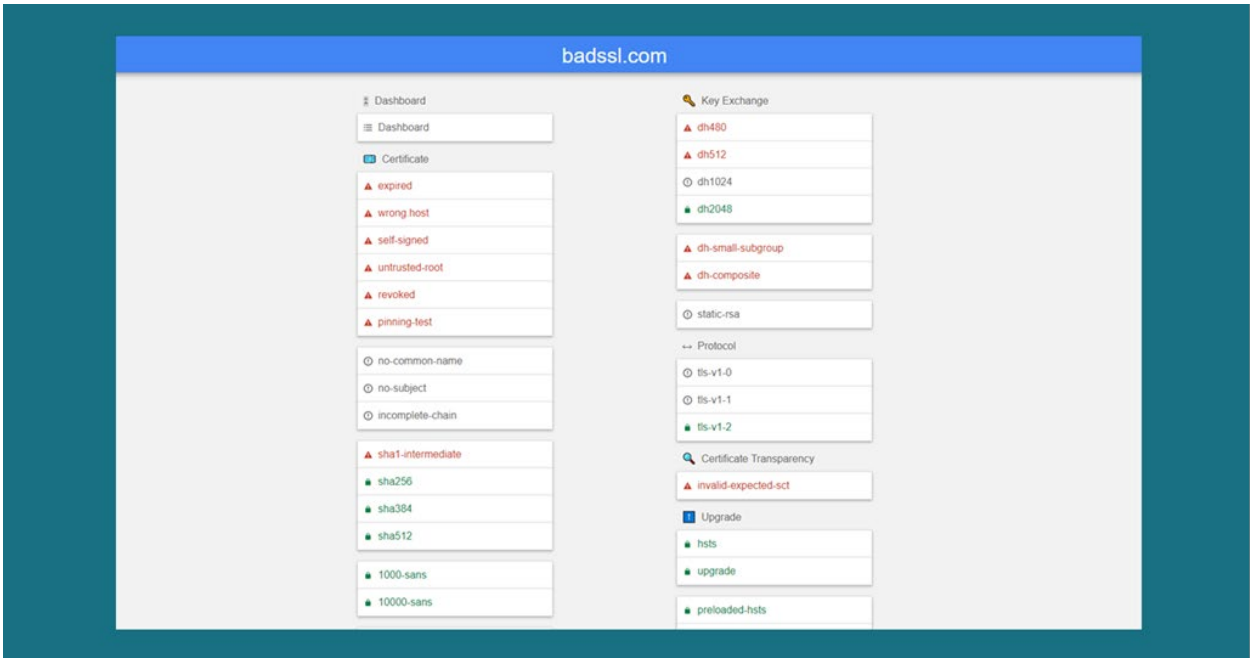
- ✓ **Google Chrome**
  “Your connection is not private.” Attackers might be trying to steal your information from “wrong.host.badssl.com” (for example, passwords, messages, or credit cards). NET::ERR_CERT_COMMON_NAME_INVALID
- ✓ **Mozilla Firefox**
  “Your connection is not secure.” The owner of “wrong.host.badssl.com” has configured their website improperly. To protect your information from being stolen, Firefox has not connected to this website.
- ✓ **Microsoft Edge**
  “This site is not secure.” This means someone’s trying to fool you or steal any info you send to the server. You should close this site immediately.
- ✓ **Safari**
  “This Connection Is Not Private.” This website may be impersonating “wrong.host.badssl.com” to steal your personal or financial information. You should close this page.

Furthermore, below are some of the reasons due to which this error occurs:

- ✓ When someone tries to access the website via an IP address, and the issued certificate is for the public-facing fully qualified domain name and not that IP address.
- ✓ The certificate is issued for test.com, but someone tries to reach the website through www.test.com. Technically speaking, WWW is a sub-domain. Though most of the time, certificates secure both WWW and non-WWW variations, there's a slim chance to experience this error.
- ✓ Generally, as it happens in shared hosting environments, multiple websites being hosted on the same IP address lead to this issue as the SSL handshake occurs before the web browser requests the host name via an HTTP header. Because of this, server fails to give enough information to the SSL/TLS Certificate resulting in generating an error. However, if you have SNI, the problem can be resolved and even a Multi-Domain SSL Certificate can also prevent from this issue.

Additionally, Name Mismatch error can be checked using an SSL Certificate Checker tool, and to resolve the same, you should know what the exact reason is. This will help solve the problem by adjusting the website's configuration.

Apart from this, there are several different types of errors such as certificate expired, wrong host, revoked, pinning test, etc. To know more about these errors, you can check out free websites like *BadSSL.*



Just go through it and click on any of the error names and you will be shown how it looks on your web-browser.

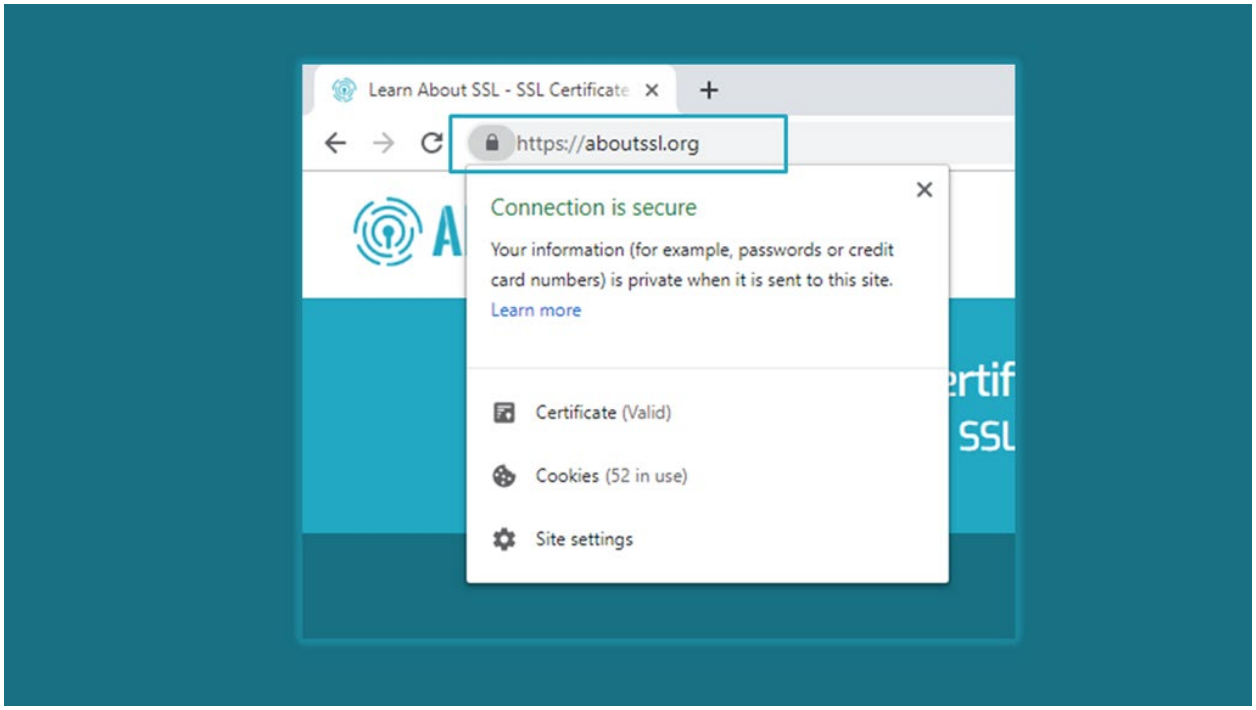
How to View SSL/TLS Certificates in Different Web Browsers

If you have come this far, you might now be well aware of how important it is for a website to have an SSL/TLS Certificate. But as a normal internet surfer, how to know if the website has an SSL/TLS Certificate installed? There must be a certain way, right? Here, we will debunk the same. Let's see how to check for an SSL Certificate on different web browsers.

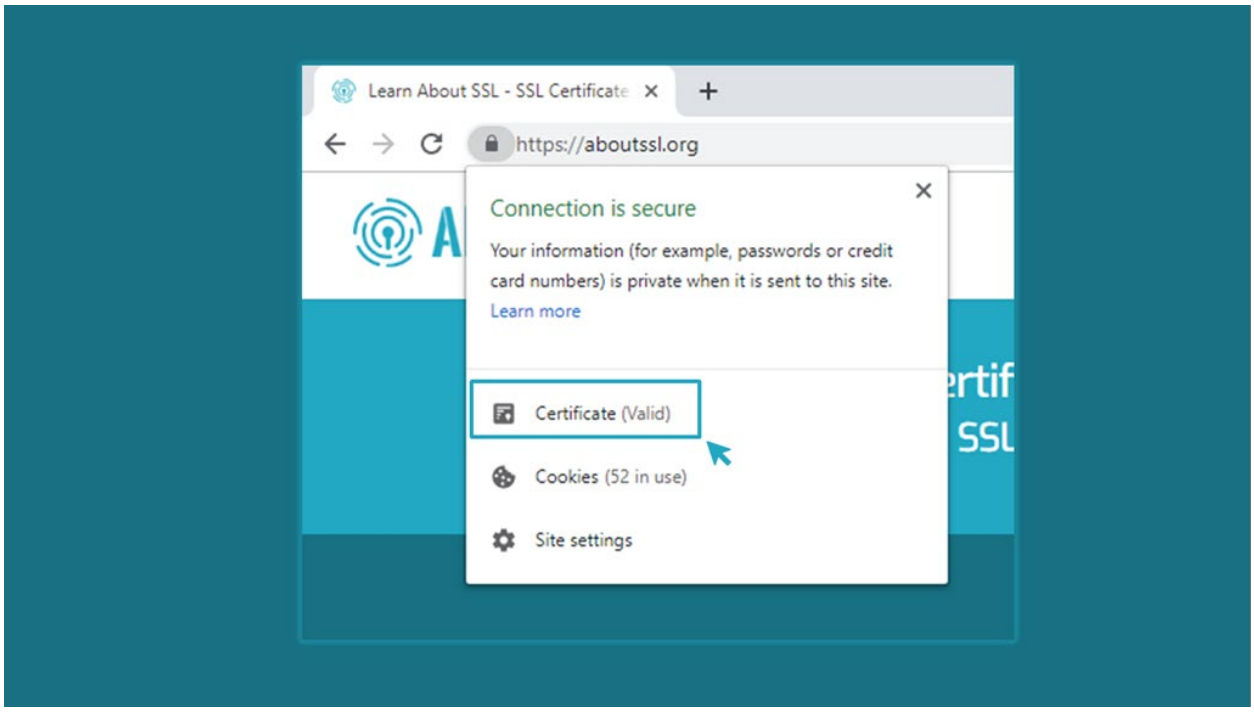
To view SSL/TLS Certificate Details in Google Chrome (Ver. 60+)

If you're not using the latest updated version of Google Chrome, then please do it. No matter which version you're using, if it's above or equal to version 61, follow the below mentioned steps:

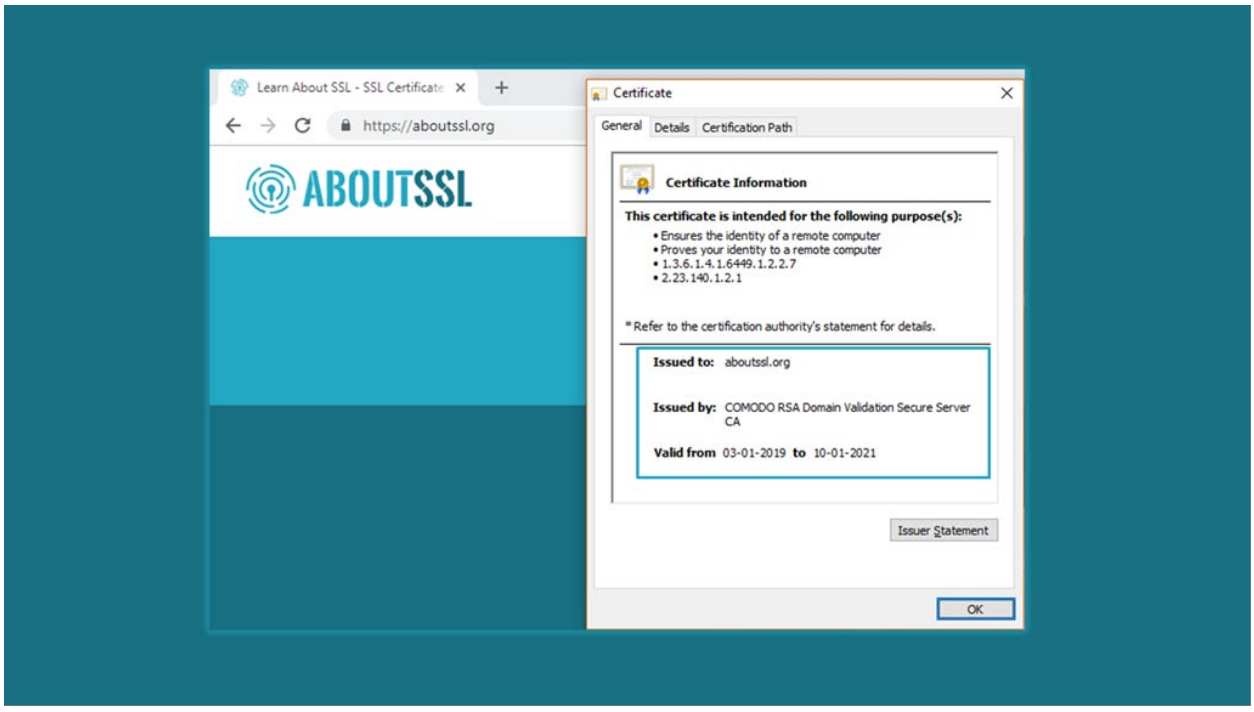
1. Go to an SSL enabled website.



2. In the address bar, click the Padlock and a popup will appear.



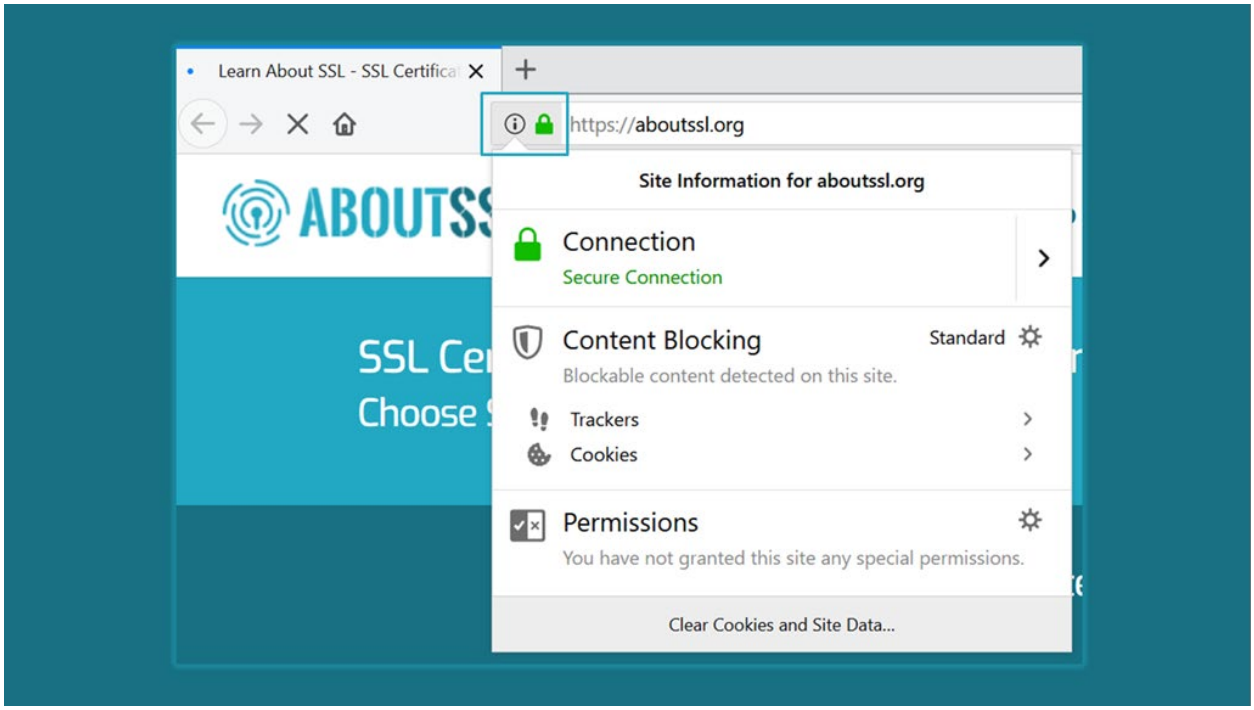
3. Now, click on the “Certificate (Valid)” option.



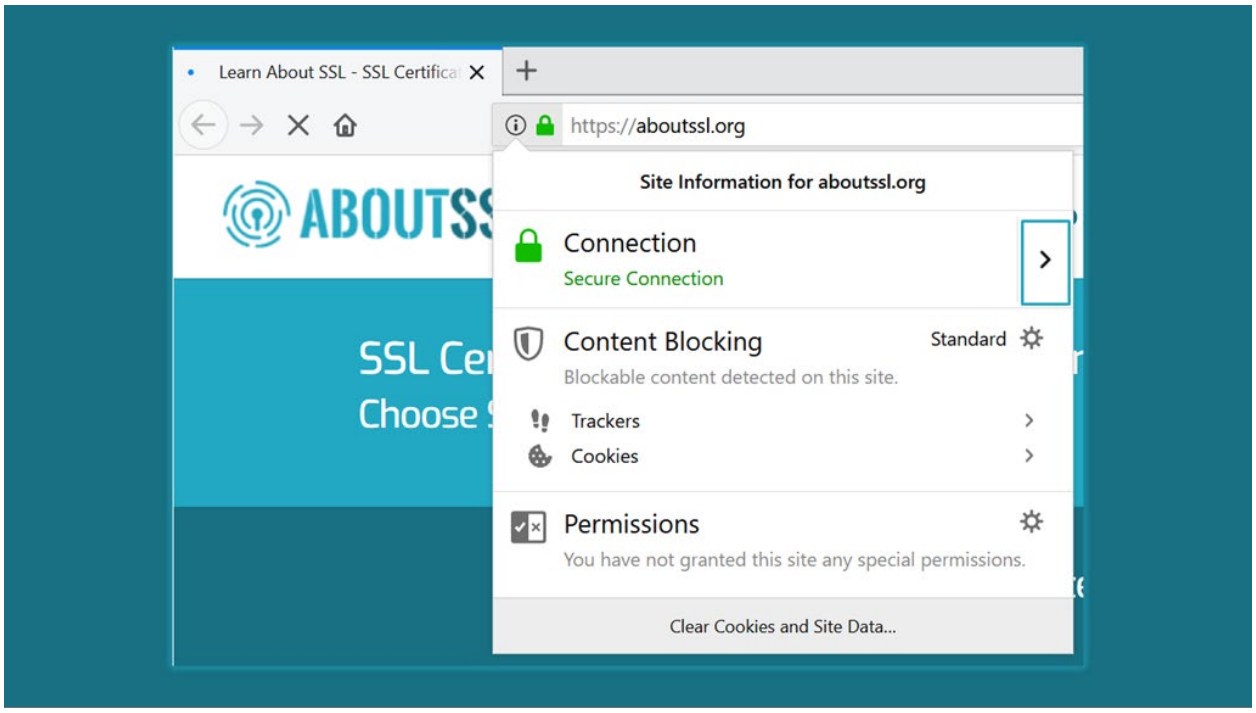
4. Once you click that “Certificate (Valid),” a certificate will be displayed.

To view SSL/TLS Certificate Details in Mozilla Firefox

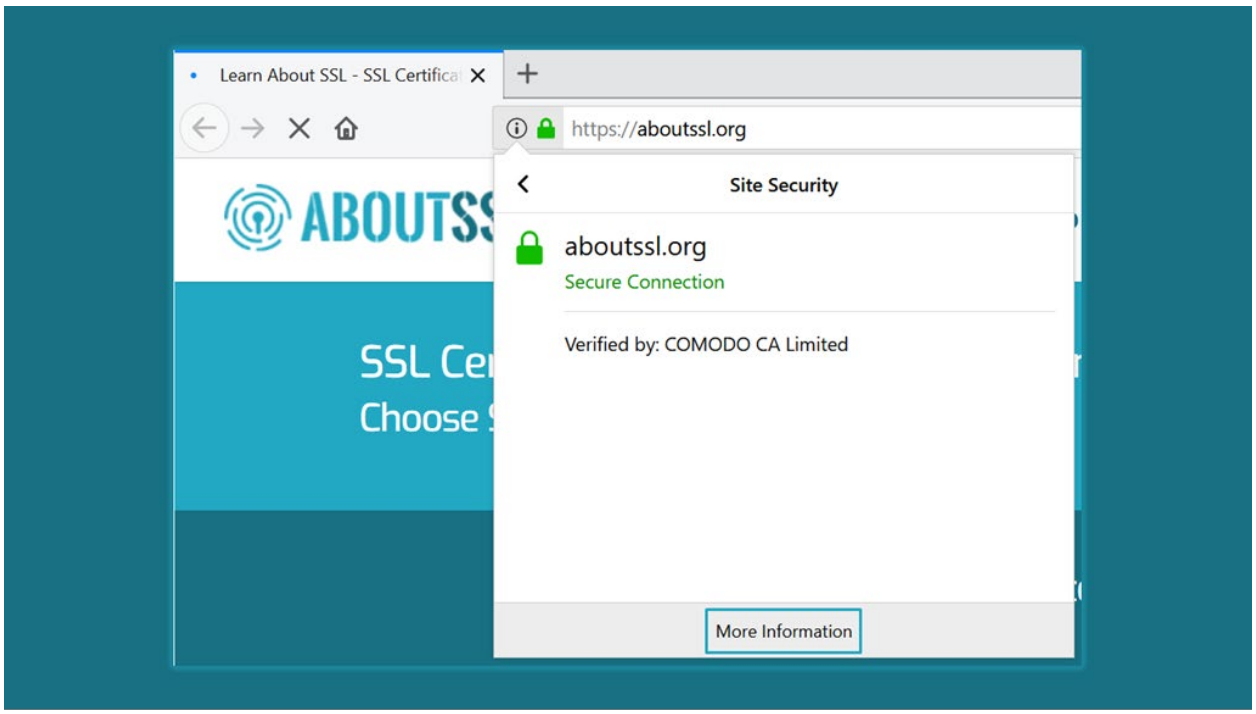
Viewing SSL Certificate details in Mozilla Firefox is quite easy. It just takes a few clicks and you're done. Follow the steps below to check SSL Certificate details.



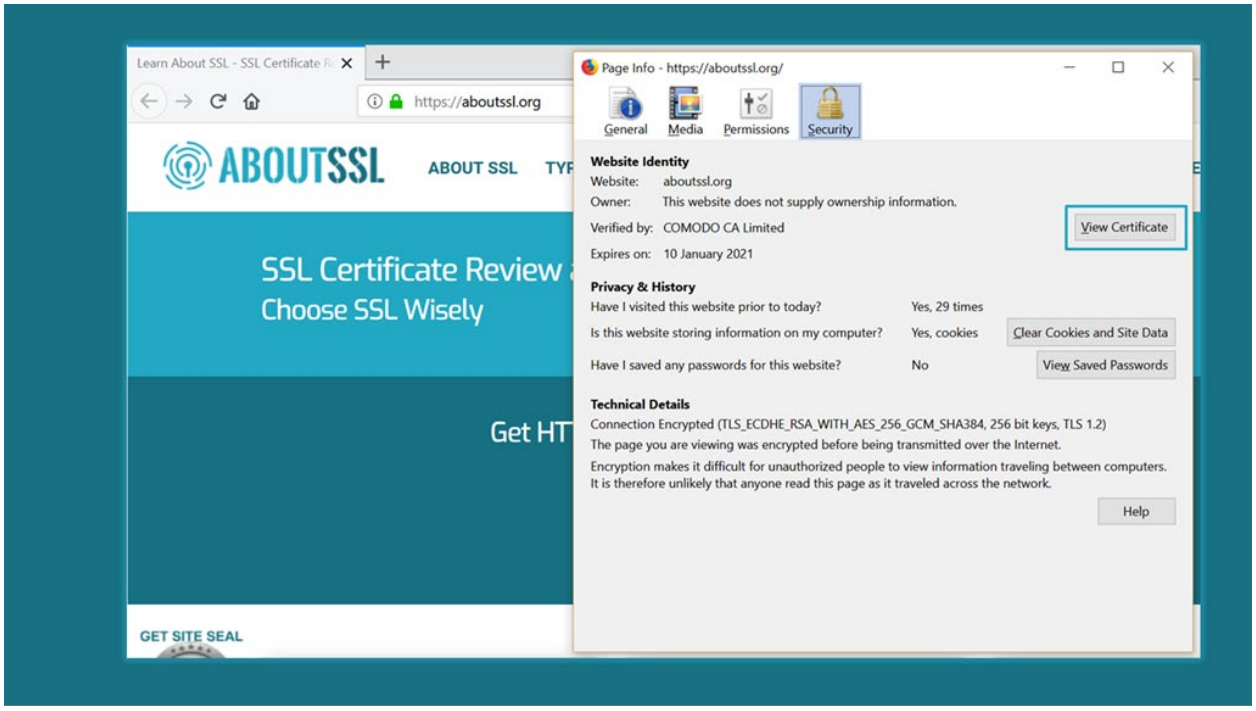
1. Go to an SSL enabled website and click on the Green padlock in the address bar.



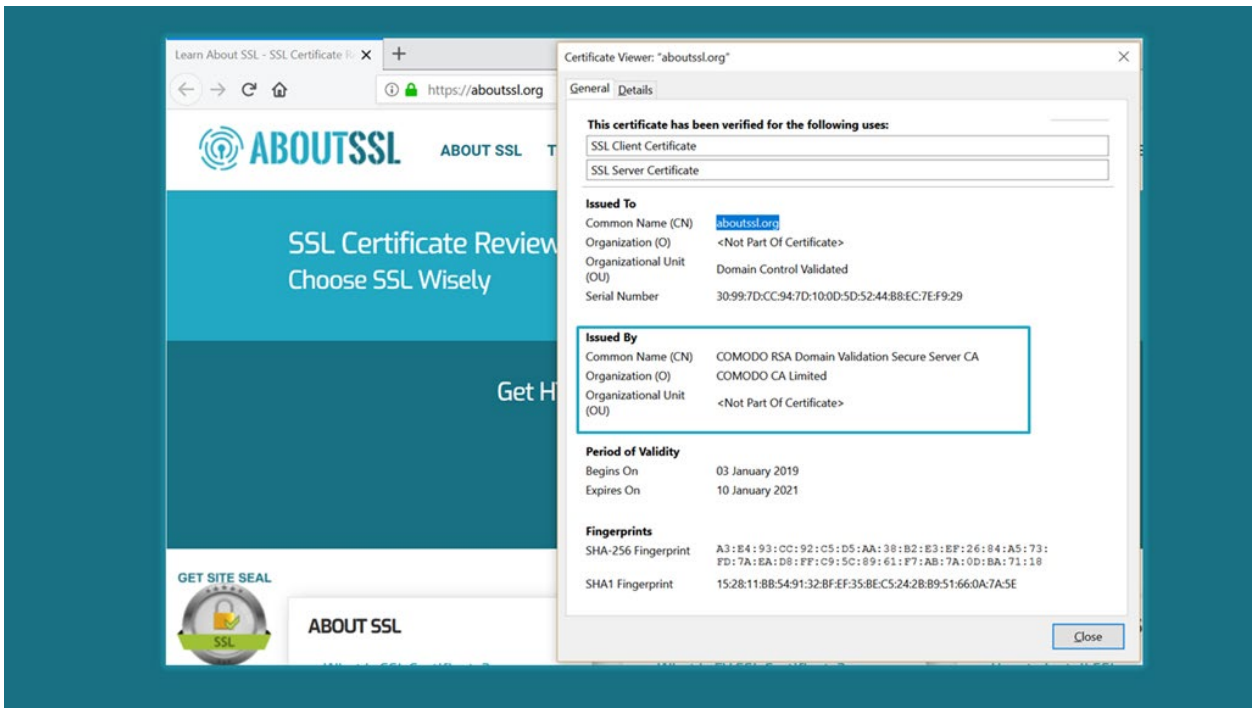
2. Click on “Connection.”



3. Click on “More Information.”



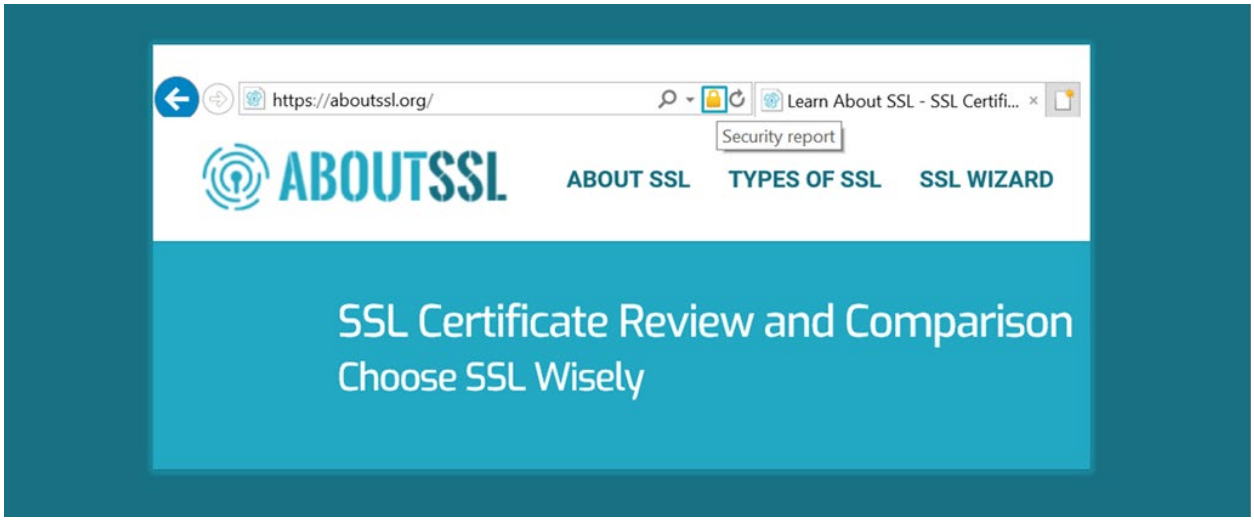
4. “Page Info” will pop up, in that, click on “View Certificate.” To know more details, click on “View Certificate.”



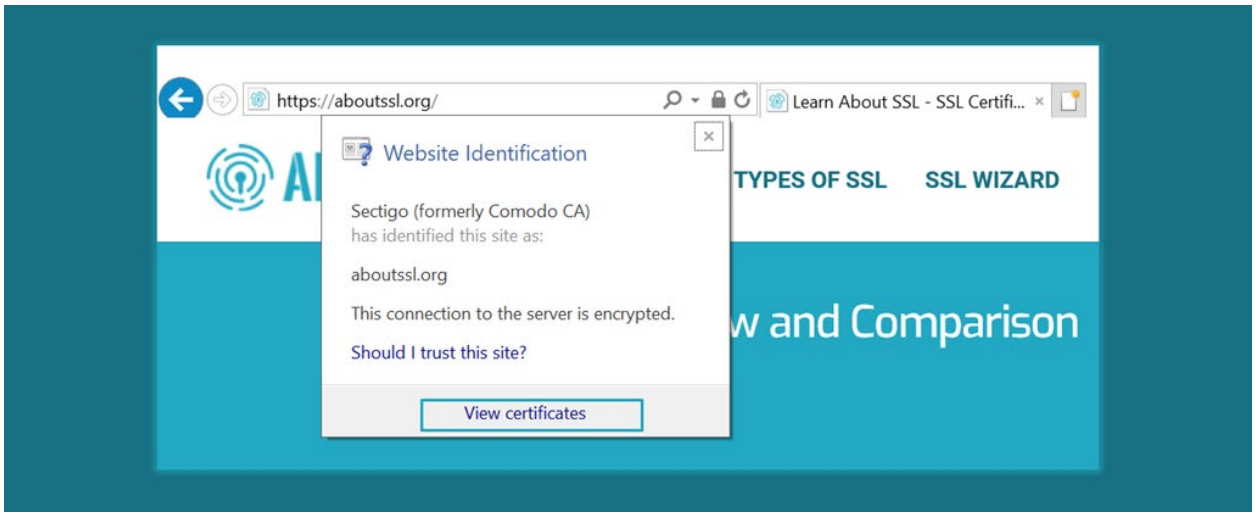
5. Once you click on “View Certificate,” a new window will pop-up which will show more details of the installed SSL Certificate.

To view SSL/TLS Certificate Details in Internet Explorer

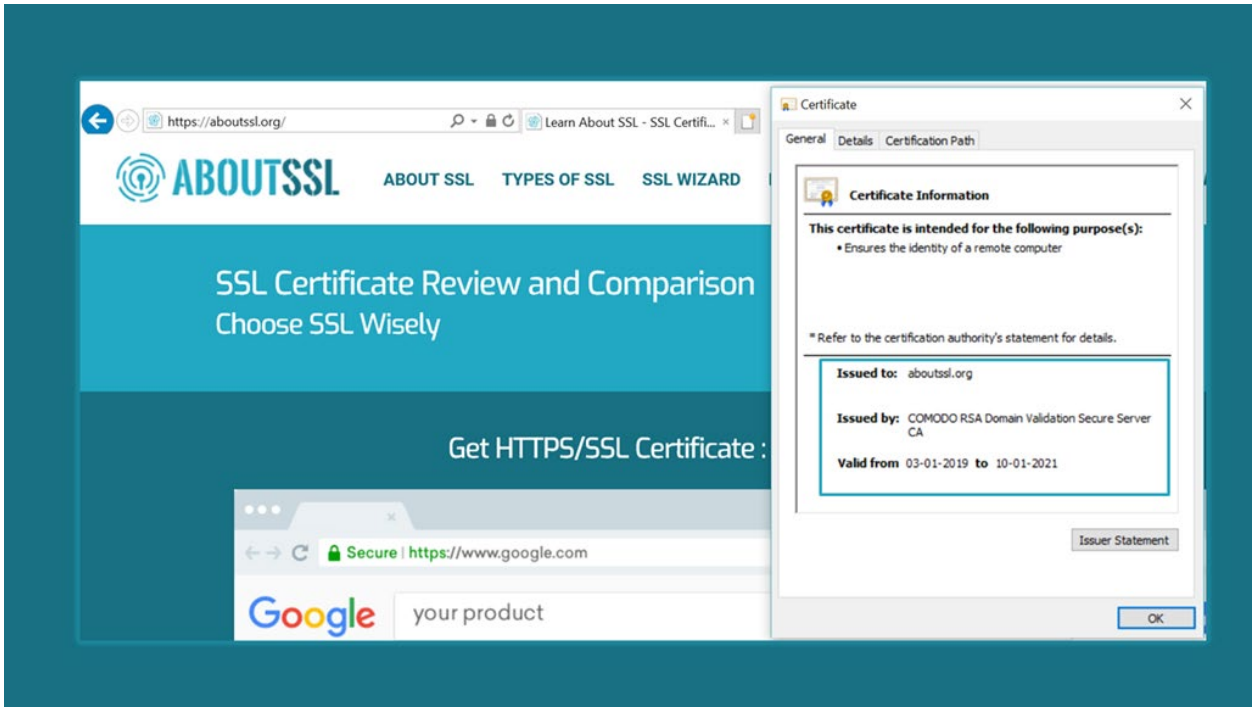
Viewing certificate details in Internet Explorer is as simple as any other web browser. Just visit an SSL Enabled website and follow the steps below.



1. Click on the padlock on the right side of the address bar.



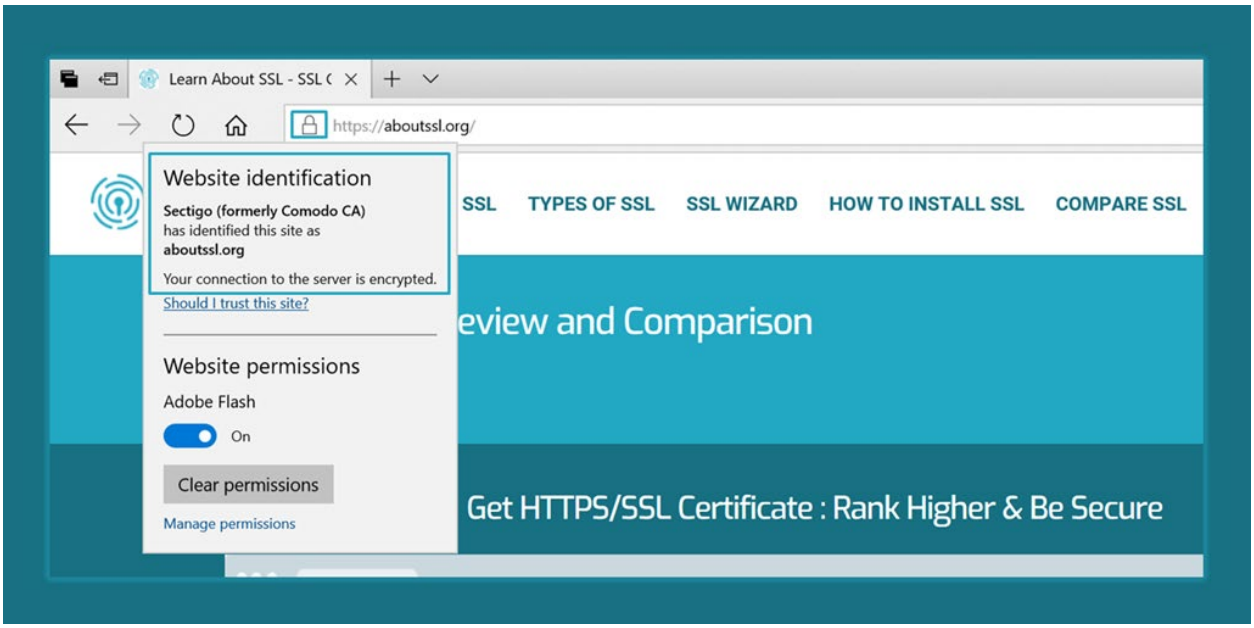
2. The “View Certificates” pop-up will appear.



3. Certificate details will pop-up.

To view SSL/TLS Certificate Details in Microsoft Edge

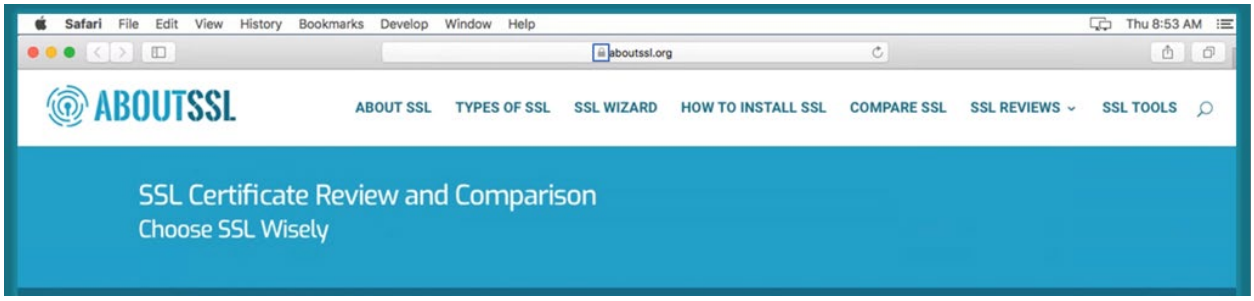
It's a simple one-way step. Visit an SSL enabled website and click on the padlock in the address bar, and the details will be displayed as below:



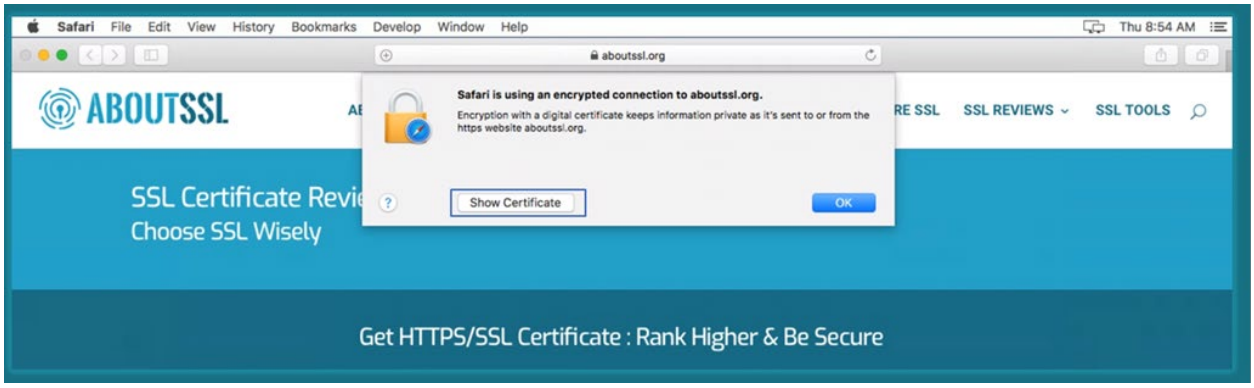
To view SSL/TLS Certificate Details in Safari

Viewing SSL/TLS Certificates in Safari is not that hard. Go through the below simple steps, and you will be able to figure it out in no time.

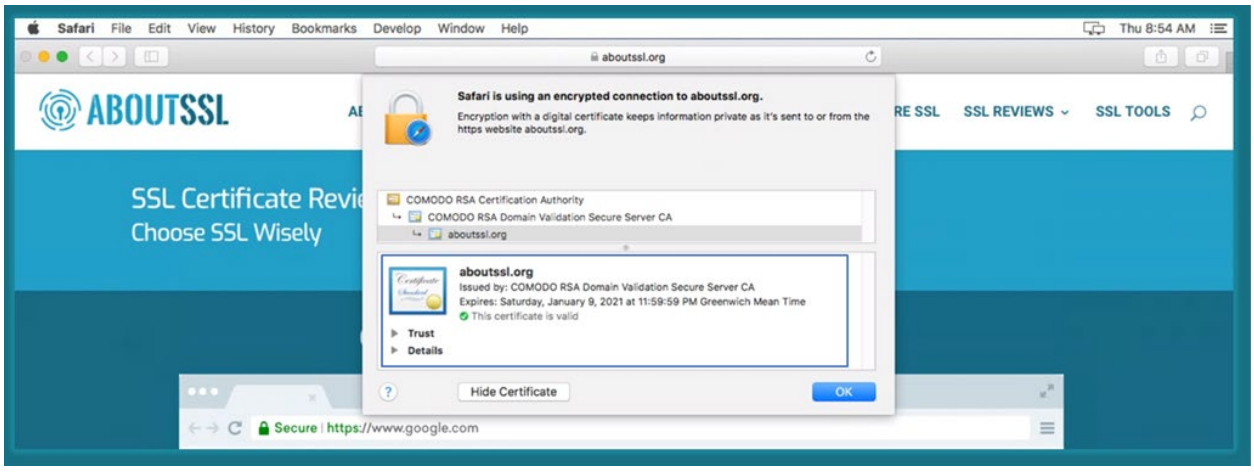
1. Open Safari and go to an SSL Enabled Website and click on the padlock, which is in the middle of the address bar.



3. Once you click on the padlock, a pop-up will open. In that, click on "Show Certificate."



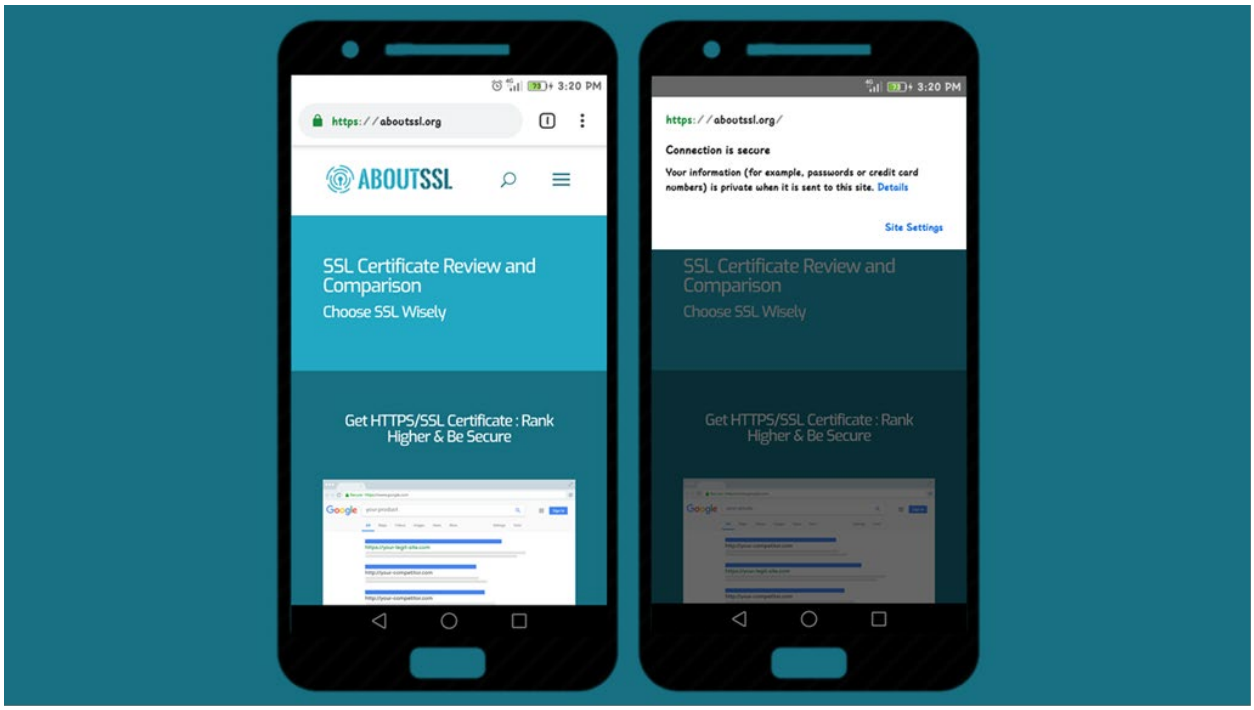
4. Finally, it will display details of the installed SSL/TLS Certificate.



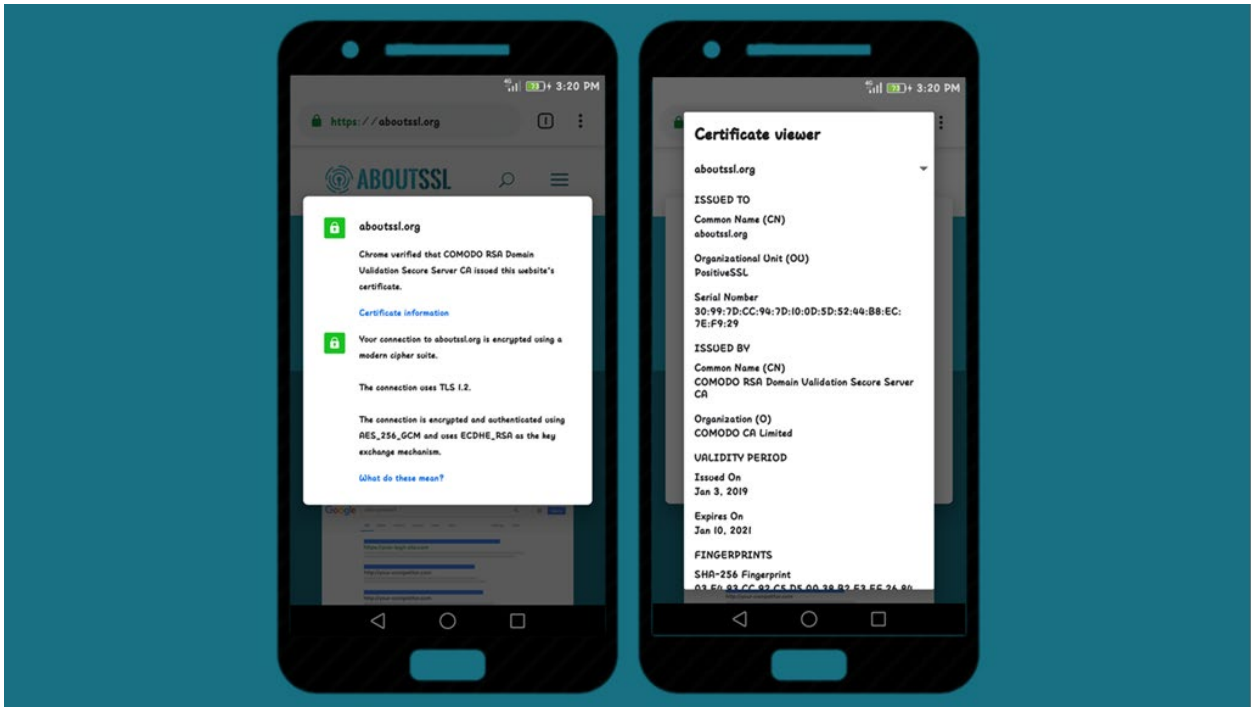
To view SSL/TLS Certificate Details in Chrome using Android Device

Viewing SSL/TLS Certificate information in Chrome using an Android device is quite similar to what we do on a desktop browser.

1. Go to an SSL enabled website and tap on the padlock in the address bar.
2. Once you tap on it, a window will pop up like this. On that, tap on “Details.”



3. A new window will pop-up. This will display the name of the SSL/TLS Certificate that has been installed. If you still want to get into the details, tap on “Certificate information.”

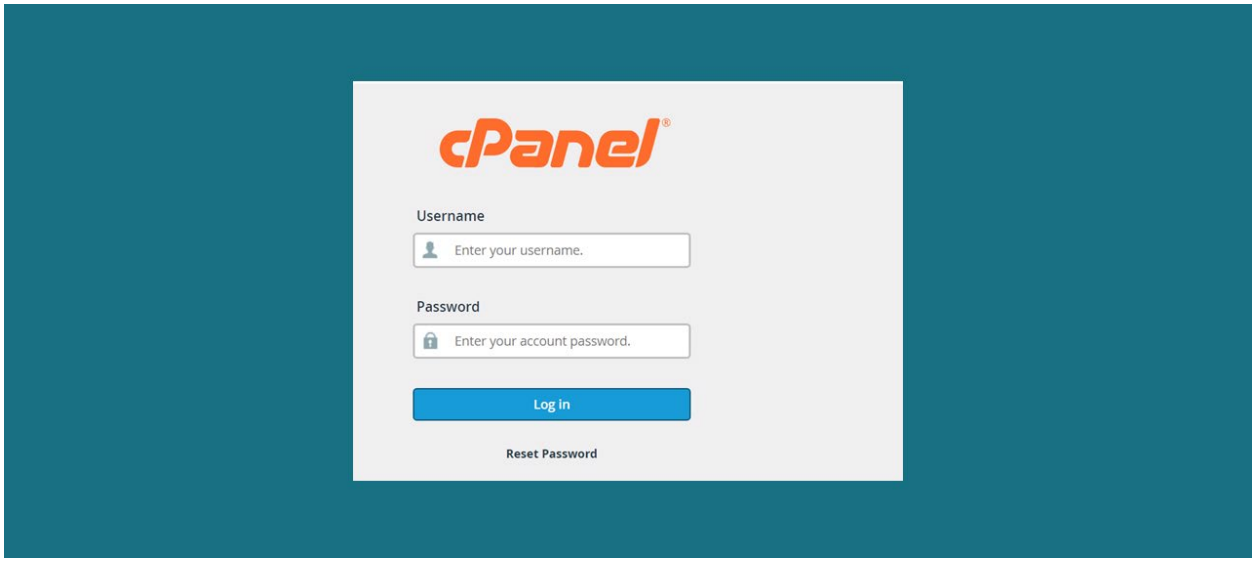


4. It will show you additional details like the Validity period and other information.

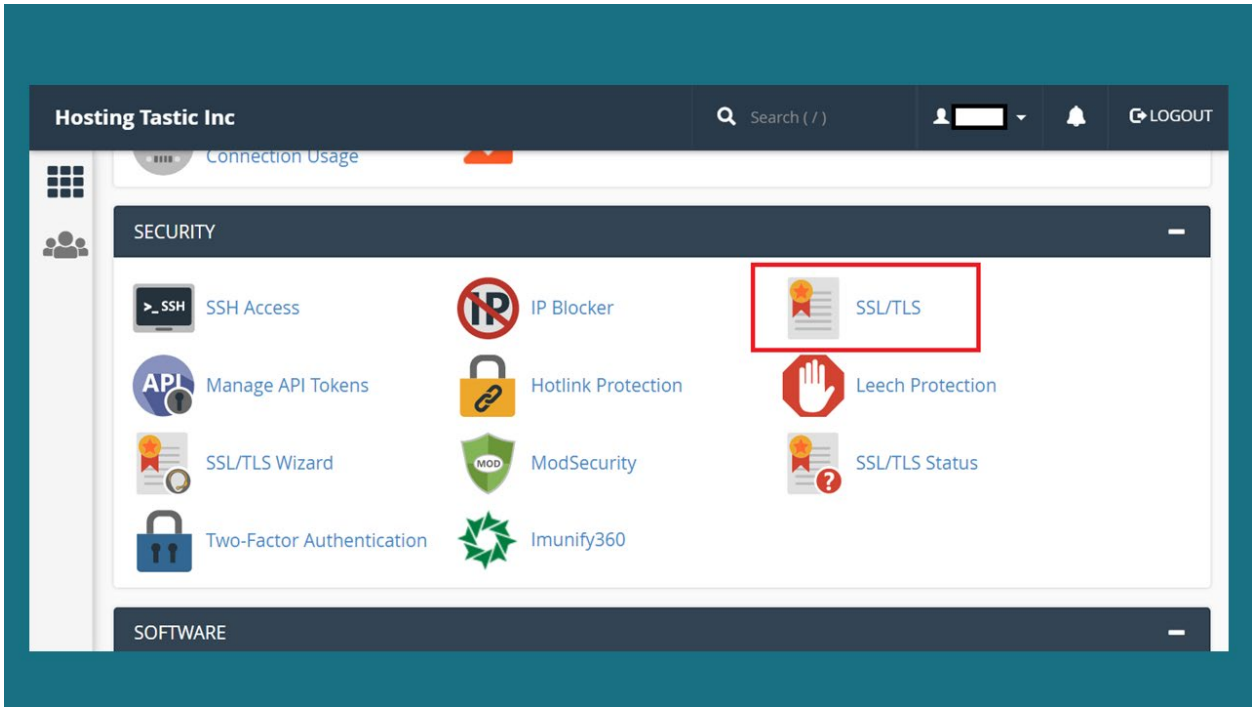
To view SSL/TLS Certificate Details in cPanel

It is quite easy to view SSL Certificate details in cPanel. It just takes a few clicks and you're done. Follow the below steps:

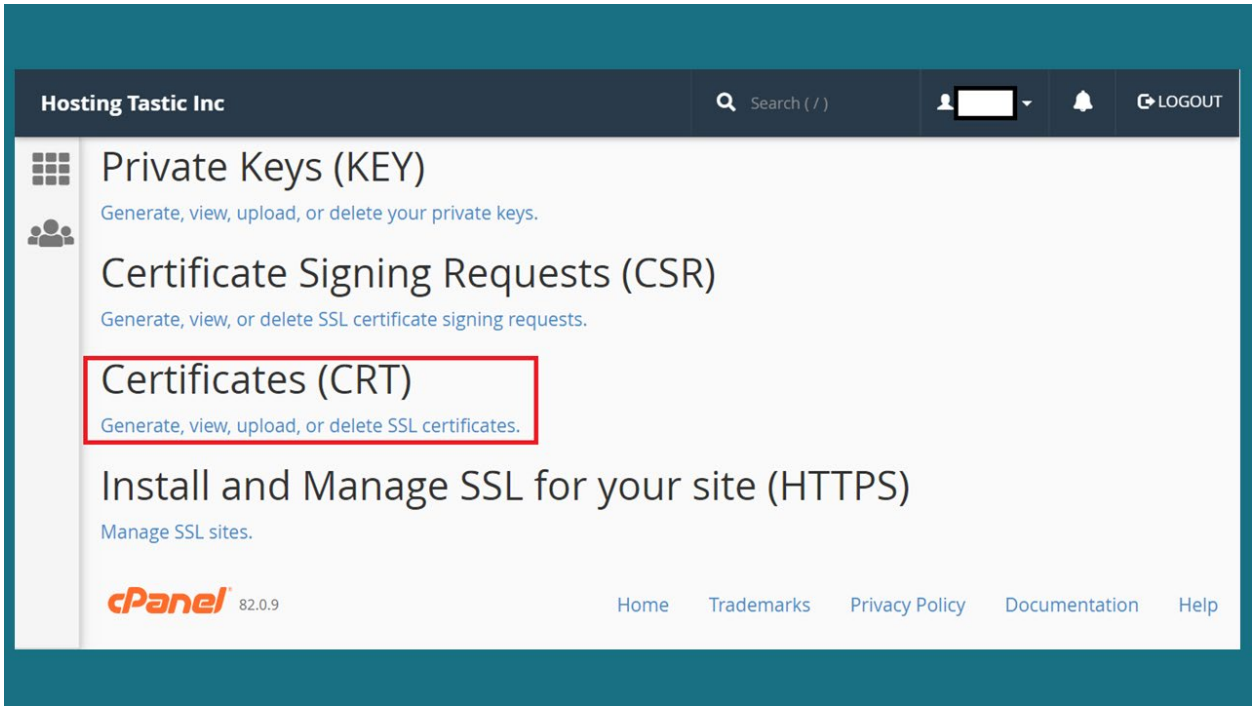
1. Login to cPanel



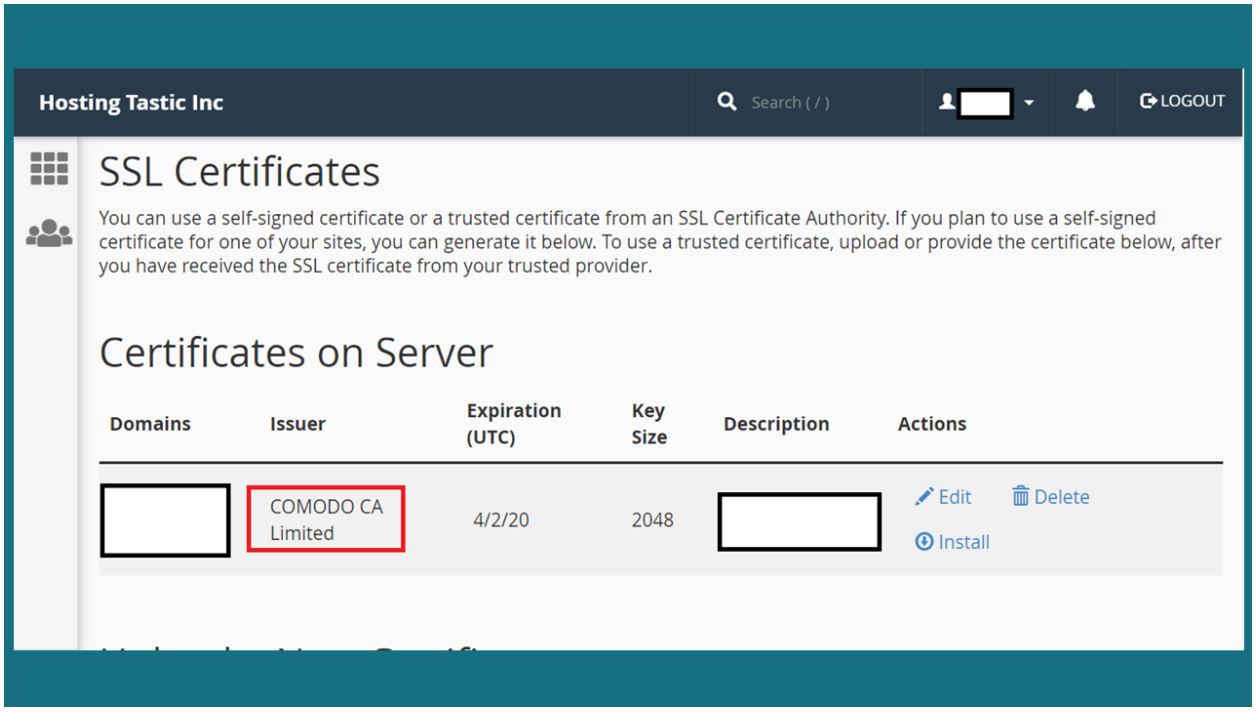
2. Go to the Security Section and Select SSL/TLS



3. Click on Generate, view, upload, or delete SSL certificates option



4. View your Installed Certificate



Appendix D: OpenSSL – An Open Source SSL/TLS Tool

OpenSSL is one of the full-featured, robust and commercial-grade software libraries for applications that are used to secure communications through the computer networks. It's one of the opensource command-line tools used by internet servers, including HTTPS websites.

It contains an opensource SSL and TLS protocol implementation, having its core library written in the C programming language. Also, it's licensed under an Apache-style license, making it completely free to get and is used for both commercial as well as non-commercial purposes, subjected with some of the conditions as per license.

Also, it does not distribute its code in binary form, but you can download it from third-party trusted websites like wiki.openssl.org. Here, you'll be provided with an option of third-party binary distributions of OpenSSL to select and download based on your platform.

Below are some of the general-purpose commands, which you may find helpful while installing SSL.

Private Key Generation Command

```
openssl genrsa -out yourdomain.key 2048
```

Command for Checking Private Key

```
openssl rsa -in privateKey.key -check
```

Command for Generating CSR

If you already have a Private Key:

```
openssl req -new -key yourdomain.key -out yourdomain.csr
```

Once this above command executed, the following additional details will be asked as below:

Country Name: 2-digit country code of where your organization is legally existing.

State/Province: Full name of the state where your organization is located.
City: Full name of the city where your organization is located.

Organization Name: Organization's legal name.
Organization Unit: Department name (Not Mandatory. Skip by pressing Enter.)

Common Name: Fully Qualified Name. For example, www.domain-name.com.

Email: The email address for processing certification. (Not Mandatory, can be skipped.)

If the Private Key is not generated:
Below command will generate both CSR as well as Private Key:

```
openssl req -new \

-newkey rsa:2048 -nodes -keyout yourdomain.key \

-out yourdomain.csr \

-subj "/C=US/ST=Florida/L=Saint Petersburg/O=Your Company,

Inc./OU=IT/CN=yourdomain.com"
```

Country Name: 2-digit country code of where your organization is legally existing.
State/Province: Full name of the state where your organization is located.

City: Full name of the city where your organization is located.

Organization Name: Organization's legal name.
Organization Unit: Department name (Not Mandatory. Skip by pressing Enter.)

Common Name: Fully Qualified Name. For example, www.domain-name.com.

OpenSSL Command for Checking CSR

```
openssl req -text -noout -verify -in CSR.csr
```

OpenSSL Command to Convert Key Files and Certificate

OpenSSL Commands to convert PEM:

Converting from PEM to DER

```
openssl x509 -outform der -in certificate.pem -out certificate.der
```

Converting from PEM to P7B

```
openssl crl2pkcs7 -nocrl -certfile certificate.cer -out certificate.p7b -certfile CACert.cert
```

Converting from PEM to PFX

```
openssl pkcs12 -export -out certificate.pfx -inkey privateKey.key -in certificate.crt -certfile
CACert.crt
```

OpenSSL commands for converting DER

Converting Certificate File from DER to PEM

```
openssl x509 -inform DER -in yourdomain.der -outform PEM -out yourdomain.crt
```

Converting Private Key File from DER to PEM

```
openssl rsa -inform DER -in yourdomain_key.der -outform PEM -out yourdomain.key
```

OpenSSL commands convert P7B File

Converting from P7B to PEM

```
openssl pkcs7 -print_certs -in certificate.p7b -out certificate.cer
```

Converting from P7B to PFX

```
openssl pkcs7 -print_certs -in certificate.p7b -out certificate.cer
```

```
openssl pkcs12 -export -in certificate.cer -inkey privateKey.key -out certificate.pfx -certfile
CACert.cer
```

OpenSSL commands for converting PKCS#12 (.pfx) file

Converting Certificate File from PFX to PEM

```
openssl pkcs12 -in certificate.pfx -out certificate.cer -nodes
```

Converting Private Key File from PFX to PEM

```
openssl pkcs12 -in yourdomain.pfx -nocerts -out yourdomain.key -nodes
```

OpenSSL Command for Checking a Certificate

```
openssl x509 -in certificate.crt -text -noout
```

OpenSSL Command for Checking a PKCS#12 file (.pfx file)

openssl pkcs12 -info -in keyStore.p12

About AboutSSL.org

AboutSSL provides an all-around SSL/TLS knowledge platform to everyone. Here, we understand that SSL can be a difficult thing to deal with, especially to users from a non-technical background. To make it easier for them, we look out to provide information regarding SSL certificates as much as possible to make the user experience easier. Also, being an affiliate member of well-known SSL providers, we are able to provide discounts on SSL/TLS Certificates, so users can get the same SSL at a lower price.

COPYRIGHT NOTICE