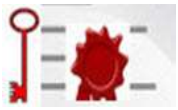
Updated:March 17, 2020 / By steve

SSL and SSL Certificates Explained For Beginners



Secure Sockets Layer (SSL) and **Transport Layer security** (TLS) are protocols that provide secure communications over a computer network or link.

They are commonly used in web browsing and email.

In this tutorial we will look:

- TLS and SSL
- Public and Private keys
- Why we need certificates and what they do
- How to get a digital certificate and understand the different common certificate types.

What is TLS

TLS is based on **SSL** and was developed as a replacement in response to known vulnerabilities in SSLv3.

SSL is the term commonly used, and today usually refers to TLS.

Security Provided

SSL/TLS provides data encryption, data integrity and authentication.

This means that when using SSL/TLS you can be confident that

- No one has read your message
- No one has changed your message

- You are communicating with the intended person (server)

When sending a message between two parties you have two problems that you need to address.

- How do you know that no one has read the message?
- How do you know that no one has changed the message?

The solutions to these problems are to:

- **Encrypt it.**– This makes the content unreadable so that to anyone viewing the message it is just gibberish.
- **Sign it**– This allows the recipient to be confident that it was you who sent the message, and that the message hasn't been changed.

Both of these processes require the use of keys.

These keys are simply numbers (128 bit being common) that are then combined with the message using a particular method, commonly known as an algorithm- e.g. RSA, to either encrypt or sign the message.

Symmetrical Keys and Public and Private Keys

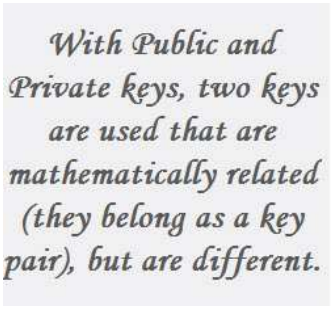
Almost all encryption methods in use today employ **public** and **private keys**.

These are considered much more secure than the old symmetrical key arrangement.

> With a symmetrical key, a key is used to encrypt or sign the message, and the **same key** is used to decrypt the message.

This is the same as the keys (door, car keys) we deal with in everyday life.

The problem with this type of key arrangement is if you lose the key anyone who finds it can unlock your door.



With **Public and Private keys,** two keys are used that are mathematically related (they belong as a **key pair**), but are different.

This means a message **encrypted with a public key** cannot be **decrypted with the same public key**.

To decrypt the message you require the **private key**.

If this type of key arrangement were used with your car. Then you could lock the car, and leave the key in the lock as the same key **cannot** unlock the car.

> *This type of key arrangement is very secure and is used in all modern encryption/signature systems.*

Hi - I'm Steve and welcome to my website where you can learn how to build IOT systems using Node-Red and MQTT.

Subscribe RSS Feed

- About Me
- MQTT Tools

Google Ads

My Youtube Channel

- node-red
- MQTT Brokers

Keys and SSL Certificates

SSL/TLS use **public and private key** system for data encryption and data Integrity.

Public keys can be made available to anyone, hence the term **public.**

Because of this there is a question of trust, specifically:

How do you know that a particular public key belongs to the person/entity that it claims.

For example, you receive a key claiming to belong to your bank.

How do you know that it does belong to your bank?

The answer is to **use a digital certificate.**

A certificate serves the same purpose as a passport does in everyday life.

A passport established a link between a photo and a person, and that link has been verified by a **trusted authority** (passport office).

A digital certificate provides a link between a **public key** and an entity (business,domain name etc) that has been verified (**signed**) by a trusted third party (A **certificate authority**)

A **digital certificate** provides a convenient way of distributing **trusted public encryption keys**.

Obtaining a Digital Certificate

You get a digital certificate from a recognized **Certificate authority (CA).** Just like you get a passport from a passport office.

In fact the procedure is very similar.

You fill out the appropriate forms add your public keys (they are just numbers) and send it/them to the certificate authority. (this is a **certificate Request**)

The certificate authority does some checks (depends on authority), and sends you back the keys enclosed in a **certificate**.

The certificate is **signed** by the **Issuing Certificate authority**, and this it what guarantees the keys.

Now when someone wants your public keys, you send them the certificate, they **verify the signature** on the certificate, and if it verifies, then they can **trust your keys**.

Example Usage

A **domain-validated certificate** (**DV**) is an **X.509 digital certificate** typically used for Transport Layer Security (TLS) where the identity of the applicant has been validated by proving some control over a DNS domain.-**Wikl**

The validation process is normally fully automated making them the cheapest form of certificate. They are ideal for use on websites like this site that provides content, and not used for sensitive data.

An **Extended Validation Certificate (EV)** is a certificate used for **HTTPS websites** and software that proves the legal entity controlling the website or software package. Obtaining an **EV certificate** requires verification of the requesting entity's identity by a certificate authority (CA).

They are generally more expensive than domain validated certificates as they involve manual validation.

Certificate Usage Restrictions- Wildcards and SANs

Generally a certificate is valid for use on a single fully qualified domain name (**FQDN**).

That is a certificate purchased for use on **www.mydomain.com** cannot be used on **mail..mydomain.com** or **www.otherdomain.com.**

However if you need to secure multiple subdomains as well as the main domain name then you can purchase a **Wildcard certificate.**

A wildcard certificate covers **all sub domains** under a particular domain name.

For example a wildcard certificate for ***.mydomain.com** can be used on:

- mail.mydomain.com
- www.mydomain.com
- ftp.mydomain.com
- etc

It can not be used to secure both **mydomain.com** and **myotherdomain.com**.

To cover several different domain names in a single certificate you must purchase a certificate with **SAN (Subject Alternative Name).**

These generally allow you to secure 4 additional domain names in addition to the main domain name. For example you could use the same certificate on:

- www.mydomain.com
- www.mydomain.org
- www.mydomain.net
- www.mydomain.co
- www.mydomain.co.uk

You can also change the domain name covered but would need have the certificate re-issued.

Why use Commercial Certificates?

It is very easy to **create you own SSL certificates** and encryption keys using free software tools.

These keys and certificates are **just as secure** as commercial ones, and can in most cases be considered even more secure.

> *Commercial certificates are necessary when you need widespread support for your certificate.*

This is because support for the major **commercial certificate authorities** is built into most web browsers, and operating systems.

If I installed my own self generated certificate on this site when you visited you would see a message like the one below telling you that the site is not trusted.

==========

Certificate Encodings and Files Extensions

Certificates can be encoded as:

- Binary files
- ASCII (base64)files

Common file extensions in use are:

- .DER
- .PEM (Privacy Enhanced Electron Mail)
- .CRT
- .CERT

Note: There is no real correlation between the file extension and encoding. That means a **.crt** file can either be a **.der** encoded file or **.pem** encoded file.

Question – How do I know if you have a **.der** or **.pem** encoded file?

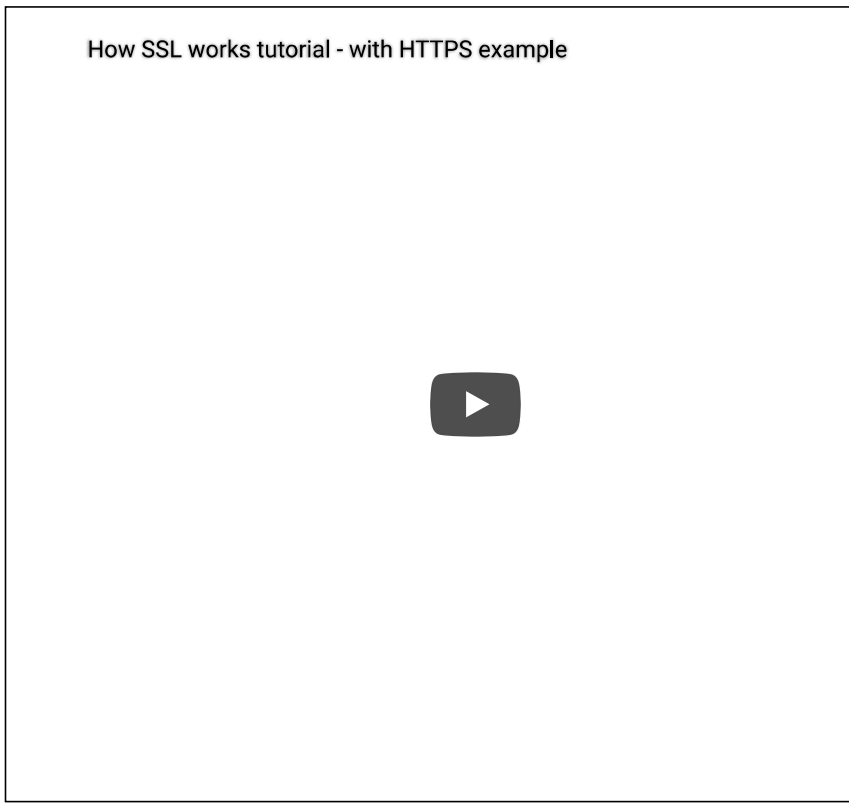
Answer- You can use **openssl tools** to find the encoding type and convert between encodings. See this tutorial – DER vs. CRT vs. CER vs. PEM Certificates

To illustrate we will look at a typical web browser and web server connection using **SSL**. (**https**).

This connection is used on the Internet to send email in Gmail etc and when doing online banking,shopping etc.

1. Browser connects to server Using SSL (https)
2. Server Responds with Server Certificate containing the public key of the web server.
3. Browser verifies the certificate by checking the signature of the CA. To do this the **CA certificate** needs to be in the browser's trusted store(See later)
4. Browser uses this Public Key to agree a **session key** with the server.
5. Web Browser and server encrypt data over the connection using the **session key**.

Here is a video that covers the above in more detail:



Digital Certificate Types

If you are trying to purchase a certificate for a website or to use for encrypting MQTT you will encounter two main types:

- Domain Validated Certificates (DVC)
- Extended validation Certificates (EVC)

The difference in the two types is the degree of trust in the certificate which comes with more rigorous validation.

The level of encryption they provide is identical

Certificate Examples

Because **.pem** encoded certificates are ASCII files they can be read using a simple text editor.

The important thing to note is that they start and end with the **Begin Certificate** and **End Certificate** lines.

Certificates can be stored in their own file or together in a single file called a **bundle**.

Root CA Bundle and Hashed Certificates

Although root certificates exist as single files they can also be combined into a bundle.

On Debian based Linux systems these root certificates are stored in the **/etc/ssl/certs** folder along with a file called **ca-certificates.crt**.

This file is a bundle of all the root certificates on the system .

It is created by the system and can be updated if new certificates are added using the **update-ca-certificates** command. See here

The **ca-certifcates.crt** file looks like this

Certificate 1
==========
-----BEGIN CERTIFICATE-----

-----END CERTIFICATE-----

Certificate2
==========
-----BEGIN CERTIFICATE-----

-----END CERTIFICATE-----

Certificate 3
==========
-----BEGIN CERTIFICATE-----

-----END CERTIFICATE-----

The certs folder also contains each individual certificate or a symbolic link to the certificate along with a hash.

The hash files are created by the c_rehash command and are used when a directory is specified, and not a file.For example the **mosquitto_pub** tool can be run as:

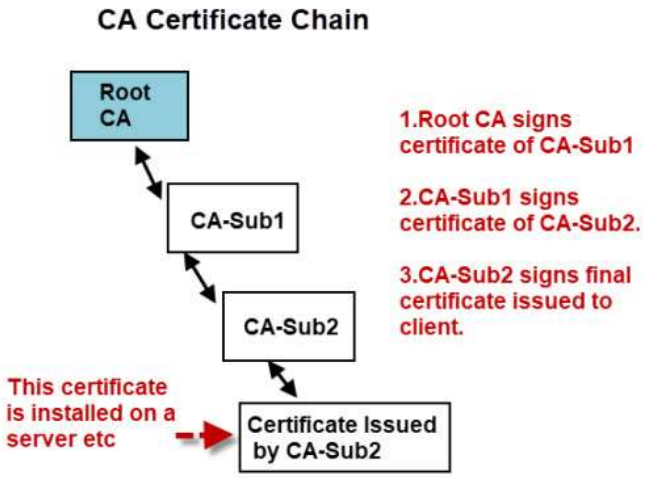
```
mosquitto_pub --cafile /etc/ssl/certs/ca-certificates.crt

or

mosquitto_pub --capath /etc/ssl/certs/
```

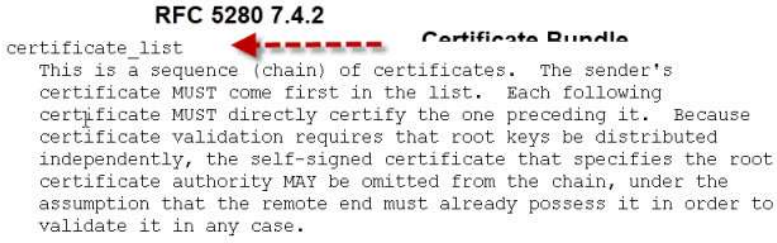
Root Certificates, Intermediate Certificates and Certificate Chains and Bundles.

A certificate authority can create subordinate certificate authorities that are responsible for issuing certificates to clients.



For a client to verify the authenticity of the certificate it needs to be able to verify the signatures of all the CAs in the chain this means that the client needs access to the certificates of all of the CAs in the chain.

The client may already have the root certificate installed, but probably not the certificates of the intermediate CAs.



Therefore certificates are often provided as part of a **certificate bundle**.

This bundle would consist of all of the CA certificates in the chain in a single file, usually called **CA-Bundle.crt**.

If your certificates are sent individually you can create your own bundle by following the steps here.

Video

- Here is my video that covers the points above.
- Here is a Microsoft video that I found that explains the above.

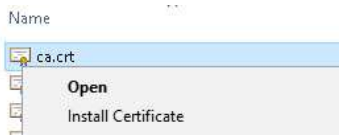
Common Questions and Answers

Q- What is a trusted store?

A- It is a list of CA certificates that you trust. All web browsers come with a list of trusted CAs.

Q- Can I add my own CA to my browser trusted store?

A- Yes on Windows if you right click on the certificate you should see an install option



Q- What is a self signed certificate?

A- A self signed certificate is a certificate signed by the same entity that the certificate verifies. It is like you approving your own passport application. see wiki

Q What is a certificate fingerprint?

A- It is a hash of the actual certificate, and can be used to verify the certificate without the need to have the CA certificate installed.

This is very useful in small devices that don't have a lot of memory to store CA files.

It is also used when manually verifying a certificate.

See here for more details

Q- What happens if a server certificate gets stolen?

A- It can be revoked. There are a number of ways that a client(browser) can check if a certificate is revoked see here

Resources

- RFC 5280
- OpenSSL Guide
- Structure of a certificate– Wiki
- Wildcard certificates – Wiki
- Certificates and Encoding
- PEM, DER, pfxpkcs12 etc
- Creating Own CA and Self signed certificates
- Digital signatures and timestamps

Related Tutorials:

- Mosquitto SSL Configuration -MQTT TLS Security
- Configure Mosquitto Bridge With SSL Encryption- Examples
- MQTT Security Mechanisms

Please rate? And use Comments to let me know more

[Total: 50 Average: 4.7/5]

78 comments

Raul Cabrera says:

May 25, 2020 at 3:46 pm

Thank you for sharing this, It was very useful.

Reply

Dev Pareek says:

April 20, 2020 at 2:17 pm

Hi Steve,

Just a thought coming in my mind since I am new to this. How a client initiates a session if there is no certificate on Clients end. Let say, if I am accessing a bank website, In that case how this will work without client certificate.

Regards
Dev Pareek

Reply

steve says:
April 20, 2020 at 4:29 pm

All browsers come with CA certificates like verisign etc already installed these certificates are all you need to access your bank server.

Reply

Diego says:
May 6, 2020 at 3:56 am

in the client-bank case, the bank is not going to request a certificate from the client. Clients are validated by providing a username and password (and extra methods if two factor authentication is used). It is the client that wants to make sure that it is connected to the bank server and not an impostor. The client will request the bank certificate and validate it. As Steve mentioned in his reply, the client browser comes loaded with CA certificates. These certificates are used to validate the bank certificate.

Reply

Chitiz says:
April 8, 2020 at 6:38 am

Hi there,
I can see public key in the browser by pressing F12 and exploring the certificate, where can I find same pub key in the server(Is it encoded in the crt file?). Also I can see the private key in server.
Also found same type of notions here:
https://superuser.com/questions/620121/what-is-the-difference-between-a-certificate-and-a-key-with-respect-to-ssl

Reply

steve says:
April 8, 2020 at 5:15 pm

If you have access to the server then just open the files as they are usually plain text.
Rgds
Steve

Reply

chitiz says:
April 9, 2020 at 12:30 am

Thanks for reply Steve.

I was little bit confused about the concept here.
Lets say we are using a command like
"openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt"

We can see two files created, one certfile and another private key file. But We dont know where the pubkey file is.
Is it embedded inside the crt file or resides at other location?
I am comapairing this with creation of key pair for ssh. When using ssh-keygen there will be two files id_rsa (private) and id_rsa.pub(public).

Reply

steve says:
April 10, 2020 at 12:36 pm

If you haven't already done so take a look at this tutorial and the steps required http://www.steves-internet-guide.com/mosquitto-tls/.
You have two key pairs an encryption key pair and a signature key pair. The private signature key of the CA is used to sign the server certificate which contains the public encryption key of the server. see step 5.
Hope that helps
Rgds
Steve

Reply

Chitiz says:
April 12, 2020 at 6:33 am

Thank You Steve...

Anshuman says:
March 17, 2020 at 4:08 am

Hi Steve,
Many thanks from for such detailed explanation. It cleared many doubts of mine and I'm sure many readers must have felt same.
Although there is one grey area on server's certificate validation on client side.
Can you put more light on how server's certificate (issued by any CA) is identified on client's browser side?
(1) I know that every browser maintains a truststore (containing some data of valid CAs) to validate against. But how actually it is done in detail?
(2) Does browser contact CA in every SSL handshake to validate server's certificate

authenticity?
(3) If CA issued cert to any server, gets stolen then how man in middle attack can be stopped? I mean thief can now impersonate server's identity.

Waiting for your answers.

Reply

steve says:
March 17, 2020 at 2:20 pm

In answer to 1 the CA certificate contains all the information to verify the server certificate.
If the server cert gets stolen in can be revoked making it invalid. The browser needs to check this. See this article
https://medium.com/@alexeysamoshkin/how-ssl-certificate-revocation-is-broken-in-practice-af3b63b9cb3
rgds
steve

Reply

Siva says:
March 16, 2020 at 9:30 am

Absolutely helpful, its been a grey area for me for many eyars, thanks steve

Reply

Phil says:
February 21, 2020 at 1:38 pm

Hi Steve,
Thanks for your post.
I have a general question..
When a CSR is created on a device I understand that a key is created too which stays on the device and the request goes to the CA for signing.
If the device cannot create a CSR for some reason and the CA is used to produce a certificate on behalf of the device, then a certificate and keys are created which need to be transported back to device.
Are the keys that are created and need to be transported to the device public or private?
Does the CA create a new key pair for this purpose?
Does this method in any way compromise the CA?
Regards,
Phil

Reply

steve says:
February 21, 2020 at 6:57 pm

Hi
The certificate is public but the key is private . Yes a key pair must be created to create the csr. Take a look here.
http://www.steves-internet-guide.com/mosquitto-tls/
It covers the process of creating a csr which is the same regardless of where you do it.

Reply

Martin Yates says:
February 10, 2020 at 3:51 pm

Hi Steve
This is a great website !
My questions is – where can the private key be stored for a certificate? I think we can use the Windows key store that we see with the windows MMC?
Java has a store called CACerts – would any pvt key be stored in there?
How about in a Databse – does some software store pvt key in a DB ?

Thanks!

Reply

steve says:
February 10, 2020 at 6:31 pm

Martin
Glad you find the site helpful. I'm not an expert on this but I did find this thread which you might find useful
https://security.stackexchange.com/questions/187096/where-are-private-keys-stored
rgds
steve

Reply

Heidi says:
January 22, 2020 at 8:58 pm

Thanks for this!

Question – I am currently setting up a connection between my company and around 50-60 others, and I require each of their public certificates. I have been asking them to send me their .cer files in a zipped email or as a text file that I will change, but I am worried that bypassing the firewall like this could cause a security issue if there were to be hidden files in the zip or the .cer is actually a virus of some sort (as I believe they run automatically on your machine as soon as being unzipped/changed to .cer). Do you have any suggestions on how I can request their certificates in a different way? I am aware that you can download them from their websites, but most people I am working with don't know much about this and will not know what website/may not even have a website.

Reply

steve says:
January 23, 2020 at 11:10 am

Hi
As far as I am aware they don't run automatically so I would have them send them as txt or zipped.
rgds
steve

Reply

Riaz says:
December 11, 2019 at 6:37 pm

Thanks Steve! It's an excellent explanation, which you call for beginner but I think it clear many doubts/misunderstanding of Seniors too (at least in my case). Good detailed concepts….

Reply

Seshu Chennupati says:
December 9, 2019 at 5:43 pm

Thanks for the explanation. My mind used to go blank when ever my manager talks about certificate changes but it will not be the case anymore.

Reply

Kamee says:
December 2, 2019 at 3:56 am

Wow, all the googling I've done over the past couple years and this is the first yours has come up. I'm just a wannabe android nerd that loves learning what all the system apps, etc. do. Thank you SO much for this amazing explanation!

Reply

Kiran says:
December 3, 2019 at 2:16 pm

excellent explanation he broken all the pices of secrets for SSL and explained in a simple way….AWESOME

Reply

ani says:
November 20, 2019 at 5:52 am

Great informative article that breaks down a complex topic in easily understandable parts. One of the best I found on net so far. Thanks a ton!

So have a question – a client of ours has a middleware (some mq that is confidential to them, and we don't know) that needs to connect to our ssl enabled url for an api integration, so they need to store our certificate in their list of trusted store and asked us to email them our certificate.
– At our Linux CentOS server we have ca-bundle.crt and ca-bundle.trust.crt : which one to send?
– And within these bundles there are many certificates for other domain names that are running on same server – how can I identify which one belongs to which domain and only send the right certificate and not all? We use certbot letsencrypt.org to generate our certificates.

Many thanks.

Reply

steve says:
November 20, 2019 at 1:34 pm

Hi
Take a look at this
https://stackoverflow.com/questions/23644473/how-can-i-split-a-ca-certificate-bundle-into-separate-files
it may help
rgds
steve

Reply

Jarrad O'Brien says:
October 23, 2019 at 4:35 am

Just wanted to say thank you for writing this article =)

Reply

Zaf says:
October 17, 2019 at 3:14 am

This was a shady area for me for years, not any more.
The better you know something, the simpler your explanation.

Reply

Zaf says:
October 17, 2019 at 3:18 am

On a lighter note why is http://www.steves-internet-guide.com still 'Not Secure' 😉

Reply

steve says:
October 17, 2019 at 4:56 pm

Yes I know it should be but migrating large websites to SSL isn't easy which is why I keep putting it off.
Rgds
Steve

Reply

Bryan says:
September 17, 2019 at 4:29 pm

Awesome blog, thanks for sharing such useful information !!!

Reply

Sandra says:
September 12, 2019 at 6:57 am

Detailed and well explained (verbal and written), without confusing reader/listeners with the technical jargons. I finally understand SSL and digital certs better now.

Reply

Ashok says:
September 11, 2019 at 11:23 am

Hi Steve, this is a very informative web page, thanks for that. I have a question on – "How do you know that no one has changed the message?" – How is this achieved usually?

Reply

steve says:
September 11, 2019 at 3:15 pm

It is the signature that checks that, But it is used in addition to encryption. You see that clearly in email where you encrypt and sign an email. Both are necessary for

complete security.
See https://support.office.com/en-us/article/secure-messages-by-using-a-digital-signature-549ca2f1-a68f-4366-85fa-b3f4b5856fc6

Reply

Pavan Kumar says:
July 18, 2019 at 7:49 am

This answers most of my questions, thanks for that.
however i do have another question tho, i.e now i need to route my traffic via cloudflare so i point my domain to ip and cloudflare does not allow me to route tcp traffic as of now. So no what i can do is sign certificates with the FQDN as the ip-address of the server (which works) and tell my iot-devices to hit the ip but becomes a mess maintaining and also will not have dos protection.
now to route my traffic via a domain with a certificate what should i do??

Reply

steve says:
July 18, 2019 at 2:45 pm

Sorry confused. Can you use the ask steve page and send me a request so that I can reply via email that way you can send me a sketch of the setup.

Reply

Top Fathers Trending Online says:
July 2, 2019 at 5:19 am

That's a great explaination. I just got SSL certificate for my website The Viral Online. Visit my profile to check it out. Thanks again...

Reply

Sameer Nagwekar says:
June 20, 2019 at 11:45 pm

Good information , cleared my concepts on certificates.

Reply

Ethanwillson says:
June 6, 2019 at 11:32 am

Very useful much-needed information. Thanx for sharing it.

Reply

Victor says:

June 1, 2019 at 10:25 pm

My site is host with WIX. My registrar is GoDaddy.
Need to renew my SSL.
In relation to the SSL what are the responsabilities of the host and of the registrar when renew a SSL.

Thanks.

Reply

steve says:
June 2, 2019 at 11:20 am

Hi
Not sure what you mean by responsibility of host? But It would be better to address the questions to GoDaddy support
Rgds
Steve

Reply

Phil says:
May 24, 2019 at 3:35 pm

Thanks! Great article!

Reply

Linda Nasredin says:
May 21, 2019 at 9:42 am

I've read many articles about SSL and certificates and this is the best one so far.
The explanation is clear and it succeeds in having the right balance of high level and details, which is not easy.
One thing I would have liked to have a little bit more details about – how the public key is used to agree on the session key.
This is the heart of the key distribution solution.

Also this link at the end of the article is not useful:
https://knowledge.digicert.com/solution/SO4264.html
I've seen it before and was disappointed to see it linked here.
It explains what a certificate fingerprint is but not its purpose.
The explanation in the article – "verify the certificate without the need to have the CA" is not enough. What is being verified?

Thanks a lot.

Reply

steve says:

May 21, 2019 at 1:08 pm

Tks for the feedback. I agree with the link not being useful and I've amended it.
Rgds
Steve

Reply

Edward says:
April 23, 2019 at 12:12 pm

This at least provided me with some ground rules for why i need an SSL Certificate and why browsers will lock out whenever I generate and sign one myself. VERY new at trying to run a website and have been facing an uphill battle to get what i once thought was a simple process done. At the same time, I'm now more aware than ever why things need to be encrypted and will probably spend the next 24 hours without sleep to try and get that going. Thank you for explaining what it is i'm trying to achieve though. Before this I thought it was simply a file that put an "S" at the end of "http" that magically makes a client more secure when browsing my site. Whelp, time to dig deeper in to the rabbit hole I suppose!

Reply

Ron says:
April 16, 2019 at 7:27 pm

Thank you for sharing this information. There's one thing I'm unsure of: You say "SSL/TLS use public and private key system for data encryption and data Integrity." – to my knowledge (which is just starting to grow), the real encryption in an https-web-session is done by a symmetric key, which in turn is shared between browser and webserver using the public/private-keys of the web server. – Am I wrong? (or do I have at least some kind of half knowledge?)

Reply

steve says:
April 17, 2019 at 3:31 pm

That's correct the actual encryption uses a symmetrical key. The public and privates keys are to protect the symmetrical key when it is exchanged.
Rgds
steve

Reply

Jonathan Gossage says:
May 9, 2020 at 2:05 pm

Could you comment on the need for this level of complexity? It looks a lot to me like grandfathering where an older existing technology, namely symmetric keys, is being

protected by newer technology, asymmetric keys, and older technology actually continues to be used. It may have been the easiest way to integrate newer technology and get some security benefits from it.

Reply

steve says:
May 10, 2020 at 9:07 am

My understanding is that symmetrical keys were used as they are less processor intensive and faster than using asymmetric keys for the actual payload encryption.
rgds
Steve

Reply

Neeraj says:
April 13, 2019 at 8:48 am

Thanks a lot Steve.
Very helpful article.

Reply

Vijay Jagtap says:
March 19, 2019 at 5:15 pm

Very well explained and concise information.
Thanks

Reply

sheetal says:
March 4, 2019 at 6:54 pm

thank you Steve... very well explained

Reply

Shobhit says:
January 23, 2019 at 12:52 pm

Hi, I want to use the Client Certificate for authentication of my users (>100 users) for a single web application. I tried this implementation with "makecert", and now I want to purchase it from a CA. I have below questions if you can answer.

1. Which type of certificate I will require? e.g. Single-domain SSL or wildcart SSL?
2. I believe 2 certificates are required i.e. Root CA certificate and Client Certificate from this Root certificate with public and private keys? Am I correct?

3. For 100+ users, will I require a single Client Certificate for all or do I need a Client Certificate for each user separately?

Reply

steve says:
January 23, 2019 at 6:10 pm

Hi
I haven't implemented client certificates yet but it is on my todo list but I'll answer best I can
If you only want to use it on a single domain then you only need a single domain certificate.
If you are purchasing the certificate then the provider will provide the CA certificate.
You will need a client certificate per client.
I will try to verify the above.
You might find these articles of interest, in particular the ibm one.
https://blog.cloudboost.io/implementing-mutual-ssl-authentication-fc20ab2392b3
https://www.ibm.com/developerworks/lotus/library/ls-SSL_client_authentication/index.html

Reply

Hemant Budhaulia says:
January 4, 2019 at 2:36 pm

i want to configure mutual TLS between client and server so what are the steps included and how many certificates are required on each side .

how can I get CA certificate ,please explain in detail

Reply

steve says:
January 4, 2019 at 3:34 pm

Only one Ca certificate is required and the client and sever require a server key and certificate.
I haven't tried it yet myself what client are you using?

Reply

Tim says:
January 18, 2019 at 1:10 am

Great article, funny how you know a lot about this, yet your sites doesn't use https, so no certificate or any of what you talk about

Reply

steve says:

January 19, 2019 at 2:33 pm

Tim
Yes I know. The reason is that the site is several years old and has lots of content. Because of the way WordPress works site links use the full url and so moving to https involves editing all pages to change links. In addition the site only provides information and doesn't require users to login which makes SSL unnecessary.
I do recommend new site owners start with SSL but for existing ones then it can be areal pain to change.

Reply

Danny says:
October 25, 2018 at 6:46 am

This is the best explanation I found yet!!!
Thanks !!!

Reply

V Rajaraman says:
October 14, 2018 at 12:05 pm

Very nice article. One small correction about the need for public-private keys. You said about symmetric keys: "The problem with this type of key arrangement is if you lose the key anyone who finds it can unlock your door".
This problem exists for any key, including public key – private key. The real issue with symmetric keys are "key distribution problem". That is, both the parties have to agree upon a common key at the beginning. But how do you get your copy of the key in the first place ? Does it need another secret key ?.. This is really chicken and egg problem, which is nicely solved by the public – private key arrangement.

Reply

steve says:
October 14, 2018 at 12:55 pm

I agree that smmetrical keys are harder to distribute. The dropping the key analogy is correct as with public/private keys you can lose the public key and not suffer but you couldn't lose the private key without suffering.
Because the keys are the same in symmetical keys if any party loses the key you are in trouble.

Reply

John Parker says:
October 8, 2018 at 8:13 am

I also want to use SSL for MacBook Repair Services website and thanks for providing me such a good information.

Reply

Prathyush says:
October 6, 2018 at 7:59 am

Very useful info and serves as a good guide for beginners.

Reply

Amine says:
October 3, 2018 at 10:39 pm

Very informative, simply explained thanks.

Reply

Abhishek Teckchandani says:
August 23, 2018 at 1:47 am

Thanks for all the information , its really useful.

I have a question on top of this, I am creating a self signed certificate for my organisation and bit confused about the common name to be used

For example the domain name of my organisation is mygroup.internal, do CN name need to be exactly same as mygroup.internal or I can append any text (env name) like test.mygroup.internal or prod.mygroup.internal

I am not sure whether this can be handled by SAN or above is a valid thing (adding text in front of CN name – env name etc)

Reply

steve says:
August 23, 2018 at 7:51 am

The common name is the name that the broker is running on and that you type into the mqtt client to access it.
For Internet connected devices it would be the domain name e.g. broker.mydomain.com.
If you use it on a local test network you can usually get away with just calling it broker and not use the domain name.
The important thing is that you can reach the broker from another machine using that name.
Does that make sense?
rgds
steve

Reply

Karthik says:

August 15, 2018 at 12:57 pm

Well explained. Thanks.

Reply

Rod says:
July 21, 2018 at 11:28 pm

Best explanation of certs ever!

Thanks.

Reply

joe smith says:
July 19, 2018 at 3:53 pm

Do we really need SSL certificate for every page? I want to get an SSL certificate for my Apple Customer Support website. So please answer my question.

Reply

steve says:
July 20, 2018 at 8:21 am

You can use an SSL certificate to secure the entire site or just parts of it. To only have it on a page you would re-direct the page from http to https which then forces it to use SSL. As this is off topic a bit for this site if you have any questions use my other site http://www.build-your-website.co.uk.
rgds
steve

Reply

Archana says:
January 17, 2019 at 8:55 am

do we really require SSL certificate sites?
It depends only on what kind of data is transmitted over communication channel. Are data sensitive which will be transmitting over internet for a page or site, if yes, then you need SSL certificate.

When http request is going from client to server or server to client and data is sensitive, then we should use SSL certificate.

SSL certificate encrypts the data when it is transmitting.

Reply

Omid Aghakhani says:

July 18, 2018 at 6:54 am

Awesome!!!

Reply

K7 Antivirus Customer Support says:

July 5, 2018 at 4:02 pm

Thanks for this article, really important to me. I was researching onSSL for a while, need it for security purposes for my company. This is one of the most relevant posts I found on it. Encryption is very reliable in performing online data transactions.

Reply

Gary says:

June 14, 2018 at 11:13 am

Easy to follow, helpful article. Thanks! However, is there any more that goes on to explain how private keys are generated in the context of a given public key, and how private keys typically get used ?

Reply

steve says:

June 14, 2018 at 1:40 pm

Public and private keys are generated as a key pair using software like openssl. This tutorial shows you how to create keys and certificates for use on a MQTT server.
http://www.steves-internet-guide.com/mosquitto-tls/
The private keys are used on the server and need to be kept secured.

Reply

PL says:

June 11, 2018 at 12:26 pm

Very nice and well explained article ! Thanks !

Reply

Serghei Tricolici says:

June 6, 2018 at 9:29 am

Great article! Very indepth explanation. Thanks a ton!

Reply

Valentino Colosso says: