Security

SSL Communication

Understanding how SSL works and steps included in establishing a secure connection

What is SSL?

When we surf the internet or search for any information, it gets transmitted to our browser from a server. The browser to server communication aka client-server communication happens over HTTP protocol (**7 layer of OSI model**) . HTTP protocol by itself cannot protect our sensitive data. That's where **SSL** comes to rescue. SSL takes place in **Layer 6 OSI Model: The presentation layer.**

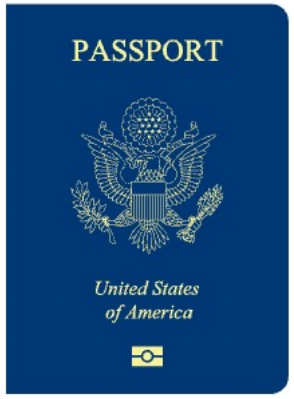
Secure Socket Layer

- SSL stands for Secure Socket Layer. It was developed by Netscape to ensure that private data remains intact when it reaches the browser.

SSL is an encryption based protocol which establishes a secure and trusted connection between browser and the server over which sensitive data can be transmitted.

- Once the connection is established, the data can be transmitted securely over the HTTP protocol. The HTTP protocol is now secured and becomes **HTTPS protocol**, which is nothing but HTTP-Secure or HTTP over SSL. To be able to establish an SSL connection, the server requires an **SSL certificate**.

Secure Socket Layer

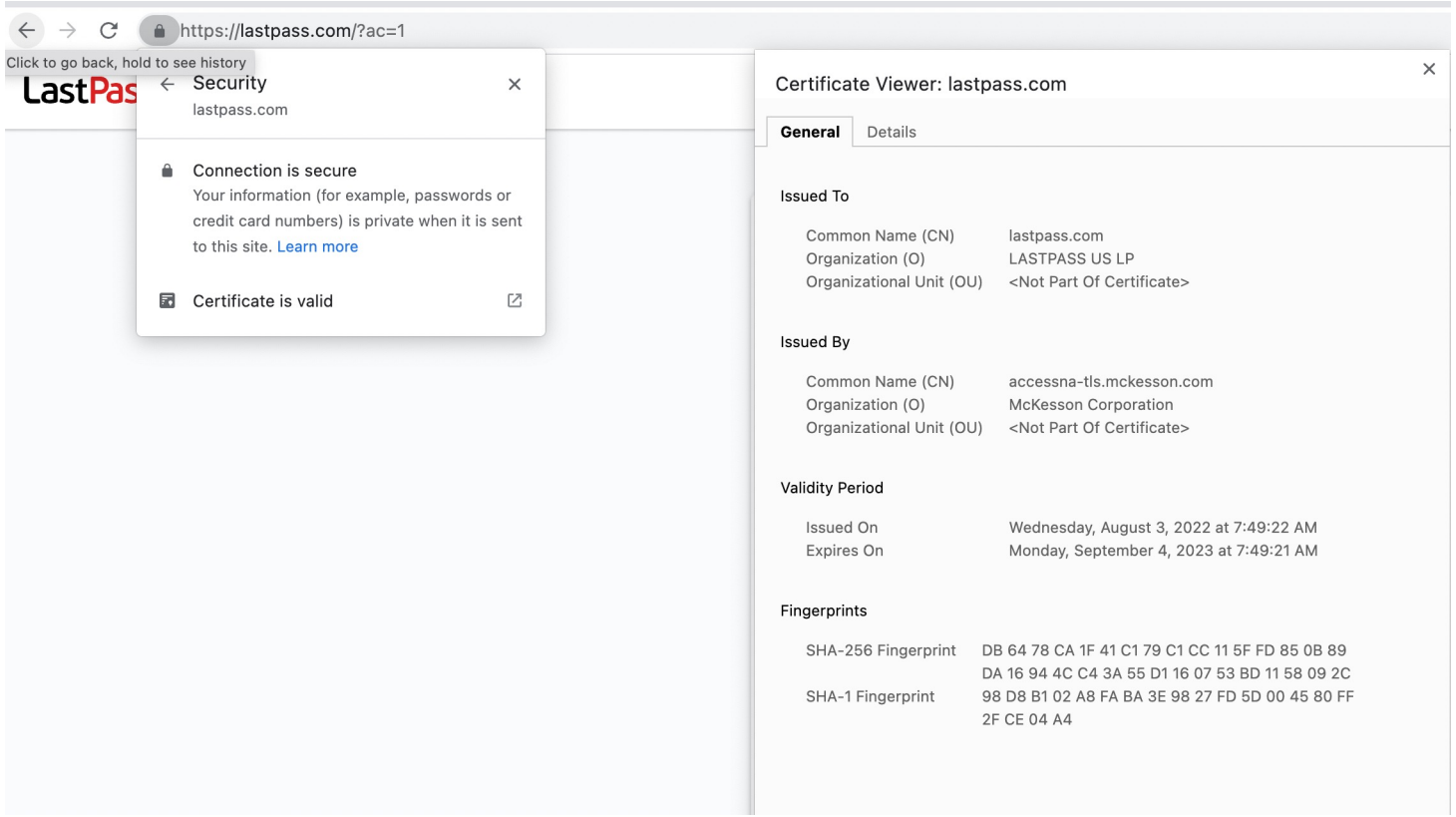


SSL certificate is an electronic document that contains a public key used to encrypt data. It also contains information about the web site which is used to prove the ownership of the public key. In simple words, it is a document that binds the encryption key with organization details.

The HTTPS certificate (also called an SSL certificate) is like an ID card for your website. It proves to the server that your website is your website.

SSL certificate

SSL certificate helps in establishing a secure connection between browser and server. It can be verified by seeing the green padlock 🔒 which appears on the address bar of the browser. We can view the certificate by clicking the padlock.

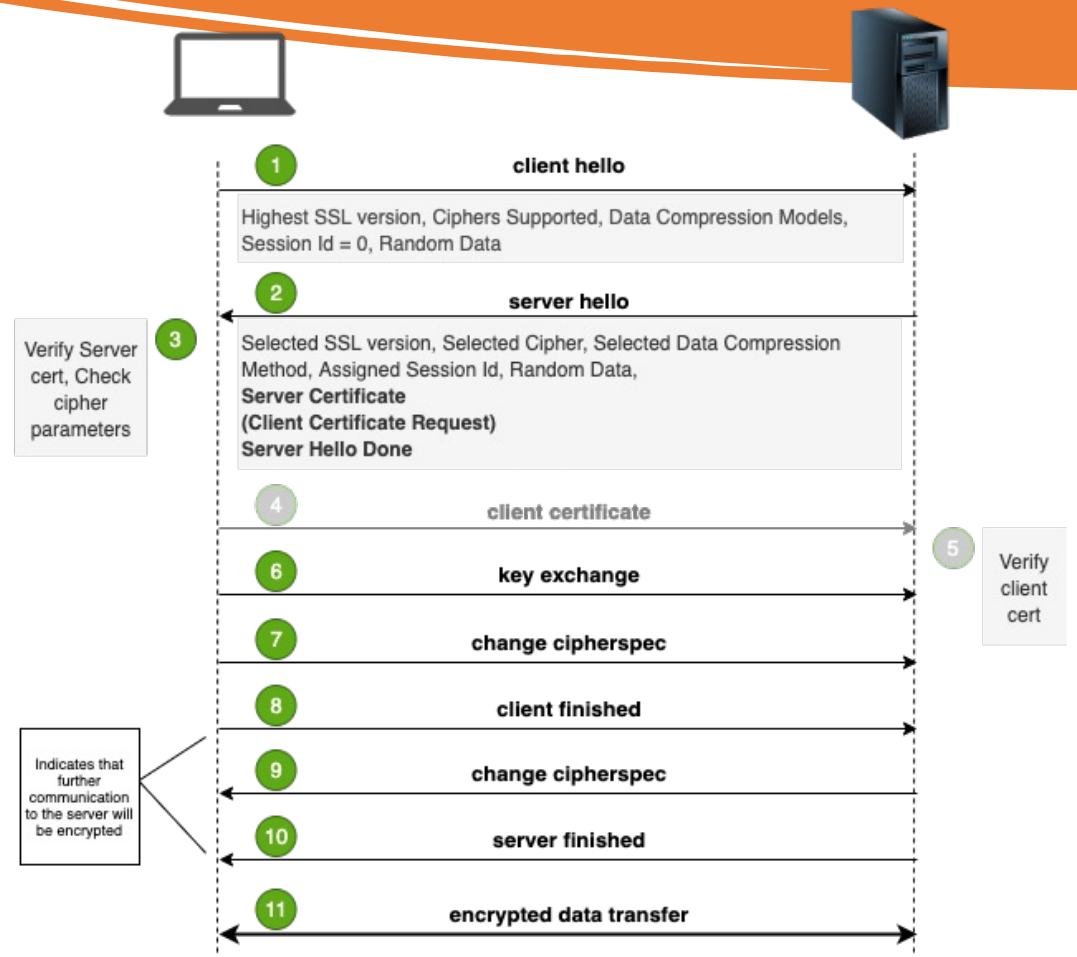


PEM certificate

These certificates are *x509* standard certificates, and the encoding of these certificates is done in two formats, **DER (Distinguished Encoding Rules) and PEM (Privacy Enhanced Mail).** In the DER format, the certificate is stored in binary form, whereas in PEM format, the certificate is stored in a human-readable text form.

```
-----BEGIN CERTIFICATE-----
MIICEjCCAXsCAg36MA0GCSqGSIb3DQEBBQUAMIGbMQswCQYDVQQGEwJKUDEOMAwG
A1UECBMFVG9reW8xEDAOBgNVBAcTB0NodW8ta3UxETAPBgNVBAoTCEZyYW5rNERE
MRgwFgYDVQQLEw9XZWJDZXJ0IFN1cHBvcnQxGDAWBgNVBAMTD0ZyYW5rNEREIFdl
YiBDQTEjMCEGCSqGSIb3DQEJARYUc3VwcG9ydEBmcmFuazRkZC5jb20wHhcNMTIw
ODIyMDUyNjU0WhcNMTcwODIxMDUyNjU0WjBKMQswCQYDVQQGEwJKUDEOMAwGA1UE
CAwFVG9reW8xETAPBgNVBAoMCEZyYW5rNEREMRgwFgYDVQQDDA93d3cuZXhhbXBs
ZS5jb20wXDANBgkqhkiG9w0BAQEFAANLADBIAkEAm/xmkHmEQrurE/0re/jeFRLl
8ZPjBop7uLHhnia7lQG/5zDtZIUC3RVpqDSwBuw/NTweGyuP+o8AG98HxqxTBwID
AQABMA0GCSqGSIb3DQEBBQUAA4GBABS2TLuBeTPmcaTaUW/LCB2NYOy8GMdzR1mx
8iBIu2H6/E2tiY3RIevV2OW61qY2/XRQg7YPxx3ffeUugX9F4J/iPnnu1zAxxyBy
2VguKv4SWjRFoRkIfIlHX0qVviMhSlNy2ioFLy7JcPZb+v3ftDGywUqcBiVDoea0
Hn+GmxZA
-----END CERTIFICATE-----
```

SSL Communication



It's like passport control in airport to allow you to be in secure zone.

The process of SSL communication is also known as an SSL handshake. It involves several steps to establish a secure connection.

Step 1
Client Hello

The client (browser) initiates the communication by sending the following details to the server:

- Highest SSL version supported
- Client Random (for generating encryption key)
- Session-Id (blank in case of new session)
- Compression Method
- Cipher Suites (most preferred at top of the list)

Step 2
Server Hello

The server replies to the client with the following information:

- SSL version selected by the server from the list provided by the client.
- Server Random
- Session-Id
- Compression Method (selected from client’s list)
- Cipher Suites (selected from client’s list)
- Server Certificate

Step 3 Client certificate

This step is optional and is used in 2-way SSL. In this step the client sends its SSL certificate to the server if the server has requested for it in step 2. In this way, the server authenticates the client.

Step 4
SSL verification

Until now the server hello is done. Now the client verifies the SSL certificate provided by the server by reading the CA (Certificate Authority) from the certificate and by loading the public key of that CA from the browser's trust store and by verifying the signature. If the certificate is not valid, the browser produces a warning, otherwise, the browser shows a green padlock at the address bar showing the authenticity of the website.

Step 5 Key exchange

This step aims at achieving a symmetric key which will be used for further communication. There are various algorithms for doing it. RSA & Diffie Hellman are two of those algorithms. The RSA algorithm uses the server's public key for confidentiality while exchanging secrets. While in the Diffie Hellman algorithm, no secret key is exchanged and the server's public key is not used. Here, the secret key of the client and the server changes for every session.

Step 6 Change Cipherspec

In this step, the client and the server have the key and now onwards the communication happens over an encrypted channel. At this step, the client and server finalizes the cipher spec. This is the last chance to change the cipher spec. After this the key exchange phase finishes.

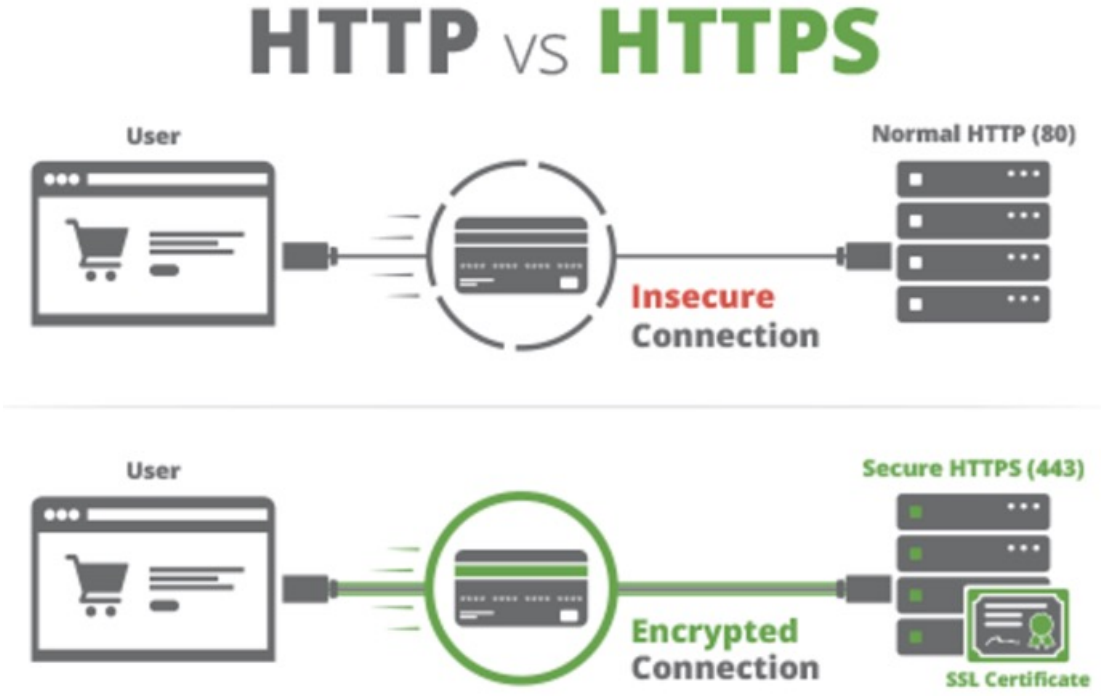
Step 7 Encrypted Data Transfer

At this step, the data which is going to be transmitted, goes through a few steps:

- The data is divided into small fragments.
- Then, these fragments are compressed.
- Then, the MAC is calculated and appended to the compressed fragment.
- Then, the symmetric encryption happens.
- Then, the SSL header is appended at the beginning of the encrypted fragment. This header tells about the SSL record type. For ex., Handshake type, data type, etc.

HTTP vs HTTPS

After all steps, the browser now has the secure and encrypted connection with the server and sensitive data can be transmitted over. This entire process is transparent and happens within a fraction of seconds



https://medium.com/swlh/demystifying-ssl-communication-5a8ea2aebcb8.

How to Obtain an HTTPS Certificate for Your Website

Steps for technical specialists (DevOps, Developers):

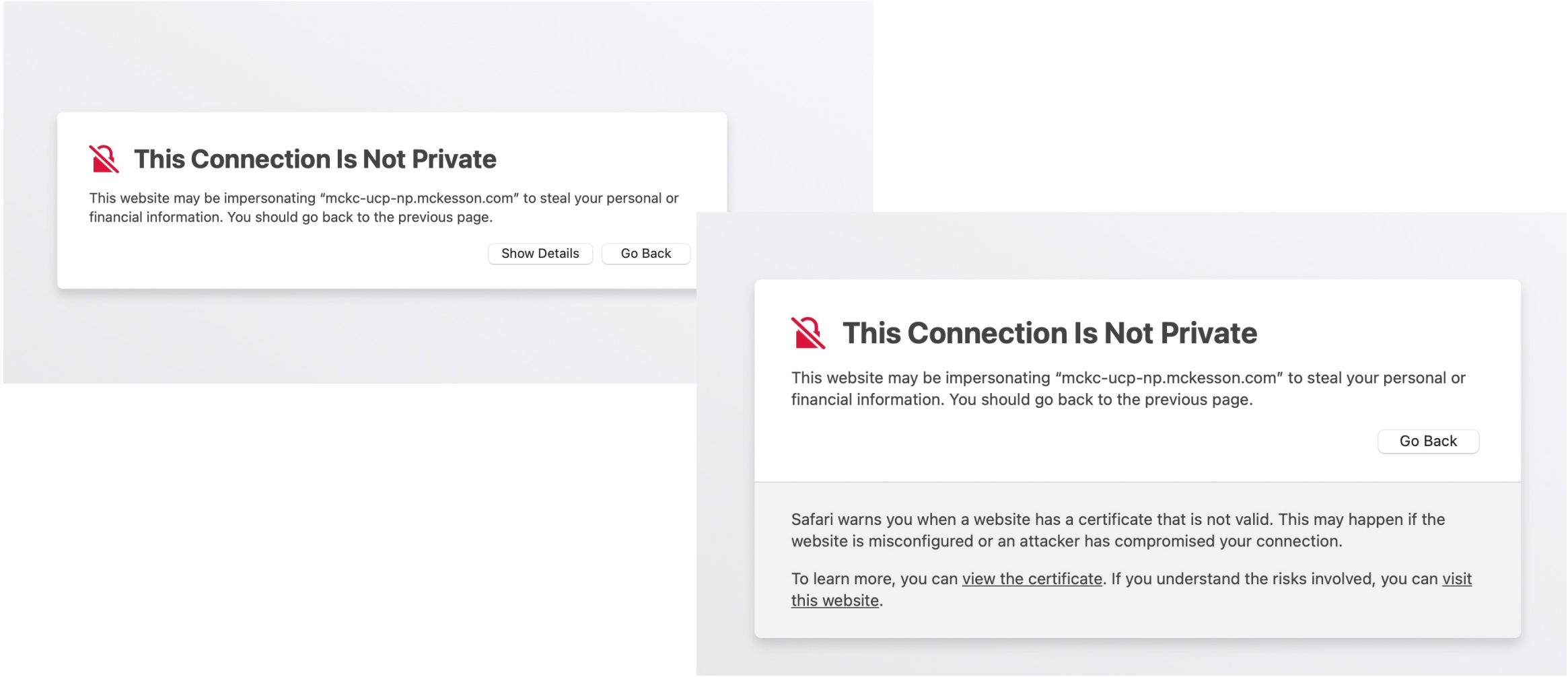
1. Switch to a dedicated IP address
2. Acquire an HTTPS certificate
3. Active the HTTPS certificate
4. Install the HTTPS certificate
5. Update your site to use HTTPS

HTTPS Certificate

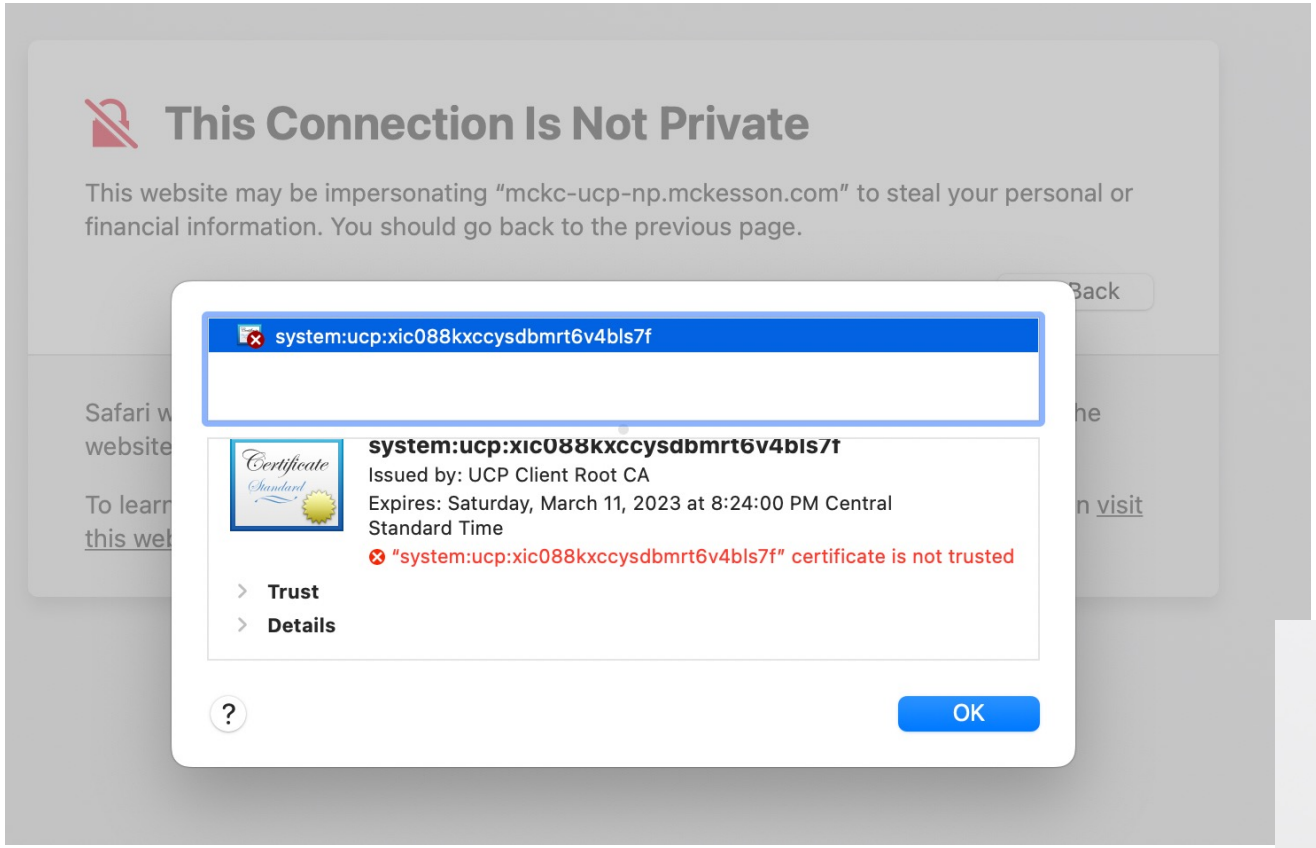
You can acquire an HTTPS certificate one of three ways:

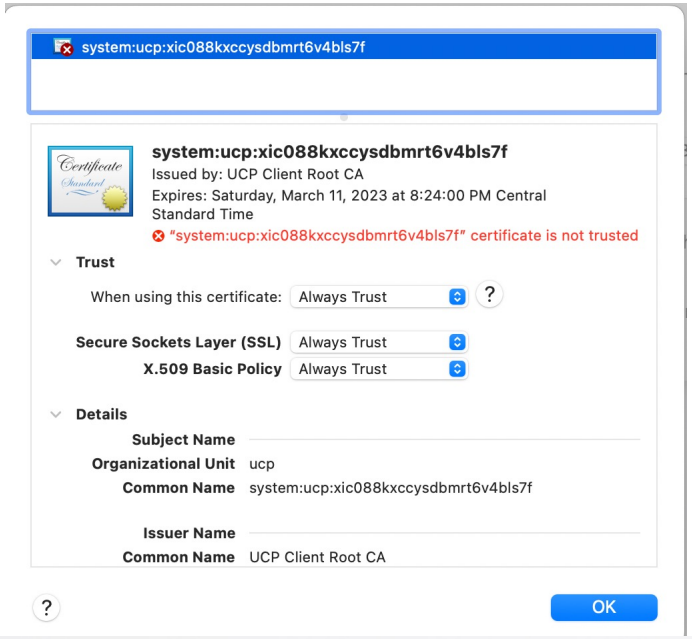
- **Paid HTTPS Certificates**: You can pay a commercial Certificate Authority (CA) or your hosting company to acquire an HTTPS certificate. For a small fee, you'll get the added benefit of technical support during installation.
- **Cloud-Based HTTPS Certificates**: Cloud providers such as Content Delivery Networks (CDNs) and Website Application Firewalls (WAFs) can also give you the benefits of a certificate. Services like Securi act as a proxy for your website, pointing your domain records to their servers and filtering out any malicious traffic.
- **Free HTTPS Certificates**: You can generate a certificate for your website on your own. Open certificate authorities such as Let's Encrypt allow you to create and install your own HTTPS certificate following their instructions.

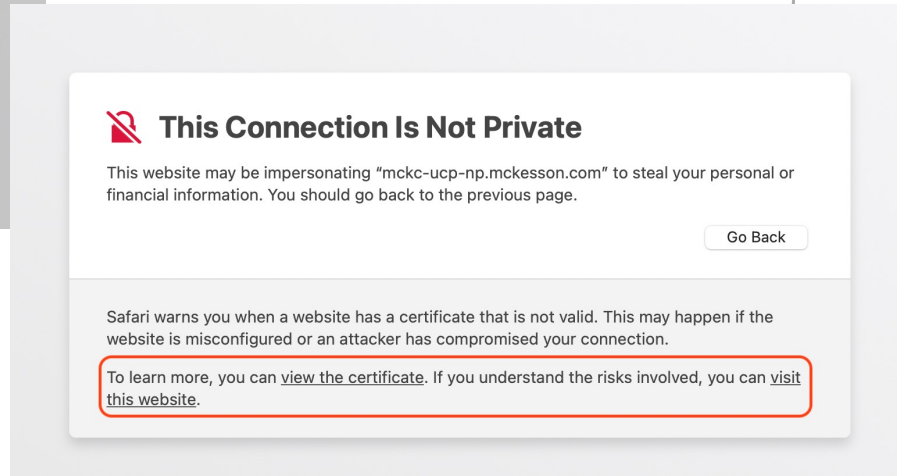
Can’t trust the certificate



Can’t trust the certificate







But you could trust the web site and visit it ->