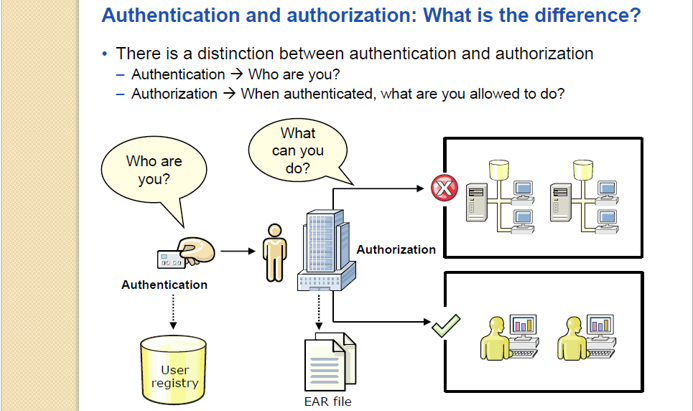
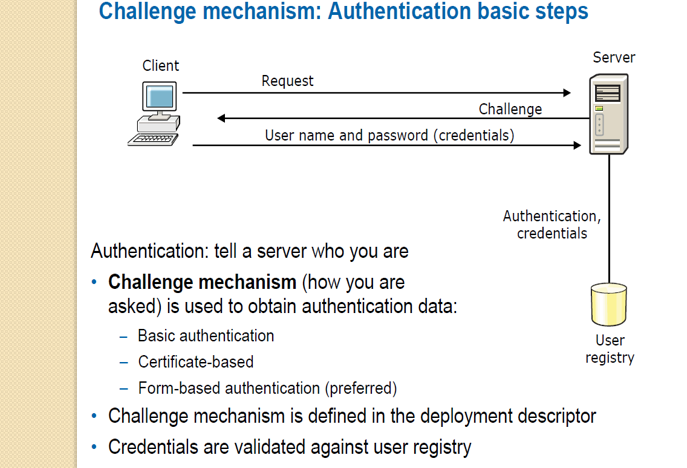
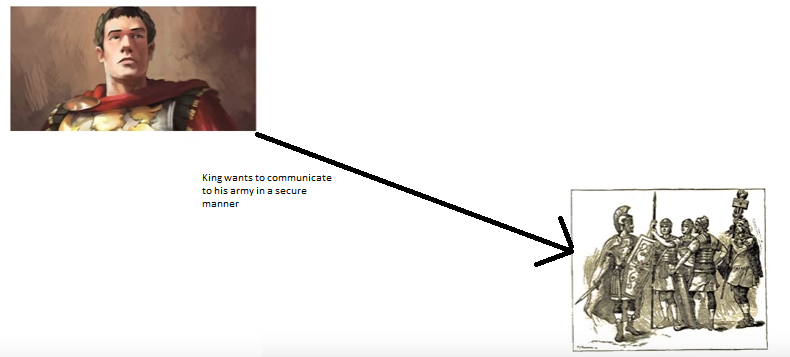
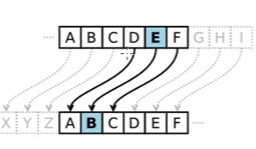
**WebSphere Security**



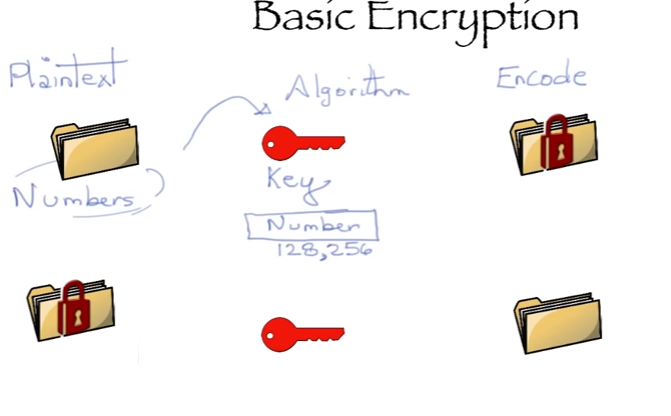




So he used encryption method- replacing the alphabets.

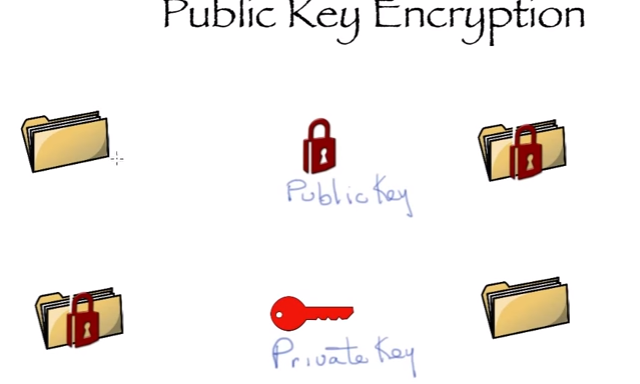


This encryption is taken care by the algorithms.

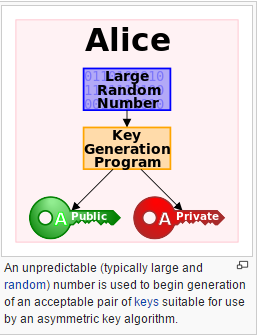


Symmetric encryption- when we are using the same key for encryption and decryption.

Public key encryption- When we are using the different key(key pair)



Key- set of large number.



How secure communication can happen with the help of key?

**Public key is for a party is available over internet, so if anyone wants to send message to any party then he can use that public key of that party for encryption, and for the decryption that party will do with the help of private key.**

Public-key cryptography, or asymmetric cryptography, is any cryptographic system that uses pairs of [keys](https://en.wikipedia.org/wiki/Cryptographic_key): *public keys* that may be disseminated widely paired with *private keys* which are known only to the owner. There are two functions that can be achieved: using a public key to authenticate that a message originated with a holder of the paired private key; or encrypting a message with a public key to ensure that only the holder of the paired private key can decrypt it.

In a public-key encryption system, any person can encrypt a message using the public key of the receiver, but such a message can be decrypted only with the receiver's private key. For this to work it must be computationally easy for a user to generate a public and private key-pair to be used for encryption and decryption. The strength of a public-key cryptography system relies on the degree of difficulty (computational impracticality) for a properly generated private key to be determined from its corresponding public key. Security then depends only on keeping the private key private, and the public key may be published without compromising security.

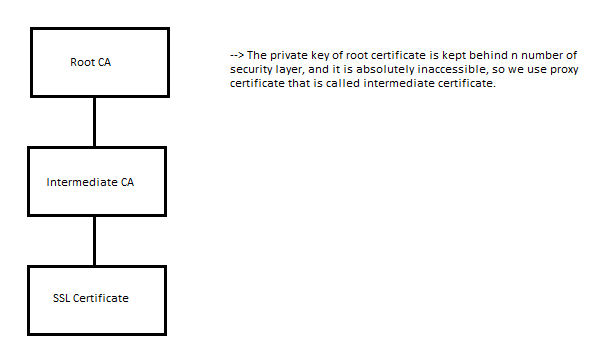
**What is an SSL Certificate?**

An SSL certificate is a bit of code on your web server that provides security for online communications. When a web browser contacts your secured website, the SSL certificate enables an encrypted connection. It’s kind of like sealing a letter in an envelope before sending it through the mail.

SSL certificates also inspire trust because each SSL certificate contains identification information. When you request an SSL certificate, a third party (such as Thawte) verifies your organization’s information and issues a unique certificate to you with that information. This is known as the authentication process.

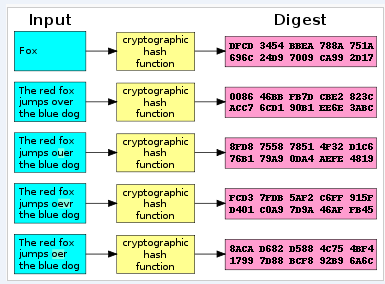
## Why does my website need an SSL Certificate?

SSL certificates keep online interactions private even though they travel across the public Internet, and they help customers gain the confidence to provide personal information on your website. If you ask users of your website to sign in, enter personal data such as credit card numbers, or view confidential information such as health benefits or financial accounts, you need to keep the data private. SSL is also used for email servers, web-based applications, server-to-server communications and more.

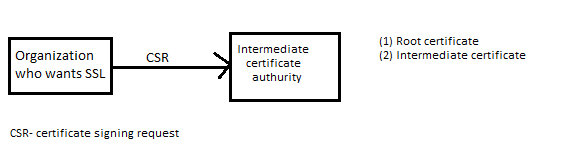
**Chain of trust**

**Hash**

A cryptographic hash function is a mathematical algorithm that maps data of arbitrary size to a bit string of a fixed size a [hash function](https://en.wikipedia.org/wiki/Hash_function).

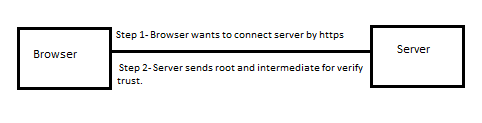


**What Happens when any organization wants security?**

****

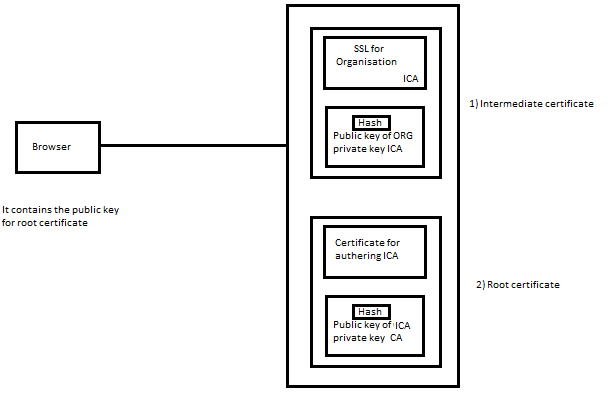
**What Happens Between the Web Browser and Server?**

1. A browser attempts to connect to a web site secured with **SSL**. The browser requests that the web server identify itself.
2. The server sends the browser a copy of its **SSL certificate**.(Root certificate and intermediate certificate)



1. The browser checks whether it trusts the SSL certificate. If so, it sends a message to the server.
2. The server sends back a digitally signed acknowledgement to start an SSL encrypted session.
3. Encrypted data is shared between the browser and the server.

**How browsers verify the trust?**



Browser is having the public key of root certificate, and as reply to server he gets root and intermediate certificate from the server.

On the basis of private key, public key and hashing algorithm it verifies the root CA, intermediate CA and certificate.

Once SSL is verified then it get public key of SSL, by using that it send encrypted message to server, and server decrypt that by using the private key and SSL communication is established.

**SSL Fundamentals**

There are 3 essential elements at work in the process described above: a protocol for communications (SSL), credentials for establishing identity (the SSL certificate), and a third party that vouches for the credentials (the certificate authority).

* Computers use protocols to allow different systems to work together. Web servers and web browsers rely on the **Secure Sockets Layer (SSL) protocol** to enable encrypted communications. The browser’s request that the server identify itself is a function of the SSL protocol.
* **Credentials** for establishing identity are common to our everyday lives: a driver’s license, a passport, a company badge. An SSL certificate is a type of digital certificate that serves as a credential in the online world. Each SSL certificate uniquely identifies a specific domain (such as thawte.com) and a web server.
* Our trust of a credential depends on our confidence in the organization that issued it. **Certificate authorities** have a variety of methods to verify information provided by individuals or organizations. Established certificate authorities, such as Thawte, are well known and trusted by browser vendors. Browsers extend that trust to digital certificates that are verified by the certificate authority.

**Lifecycle of an SSL Certificate**

If you need to secure your web site, it is quick and easy to request an SSL certificate and install it.

1. Generate a [Certificate Signing Request](https://search.thawte.com/support/ssl-digital-certificates/index?page=content&id=AR1108&ampactp=LIST) (CSR) for the web server you plan to secure. If you do not manage your own web server, contact your web host or Internet service provider to request a CSR.
2. Select an [SSL Certificate](https://www.thawte.com/ssl/index.html) and click buy.
3. Pick up your certificate in to your [Thawte® Certificate Center Account](https://ssl-certificate-center-enterprise.thawte.com/vcce/enterprise/console_login?application_locale=THAWTE_US).
4. Follow [installation instructions](https://search.thawte.com/support/ssl-digital-certificates/index?page=content&id=SO1498&actp=LIST) for your Web server.
5. Download the [Thawte® Trusted Site Seal](https://www.thawte.com/ssl/secured-seal/index.html) to display on pages within your secured domain.

At the end of the SSL certificate’s validity period (1-5 years, depending on the certificate type and your selection), you have the option to renew your SSL certificate. You may need to provide additional information for authentication or [generate a new CSR](https://search.thawte.com/support/ssl-digital-certificates/index?page=content&id=SO157).

**Which SSL Certificate is right for me?**

The right SSL certificate for your organization depends on your web site and your audience.

**What information certificate contains?**

Secure Website Certificates contain the following information.

**Serial Number**: Uniquely identifies the certificate.

**Subject**: Identifies the certificate owner, such as the name of the organization owning the certificate.

**Issuer**: Identifies the entity that issued the certificate.

**Subject Alt Name Extension**: List of website addresses that the certificate can be used to identify.

**Signature**: Data that verifies that the certificate came from the Issuer.

**Signature Algorithm**: Algorithm used to create the Signature.

**Valid-From**: The date the certificate is first valid.

**Valid-To**: The expiration date.

**Key-Usage and Extended Key Usage**: Specifies how the certificate may be used, such as for confirming ownership of a website (Web Server Authentication).

**Public Key**: The public part of the data that comprises the public/private key pair. The public and private keys are mathematically linked, so the data encrypted with the public key can only be decrypted with the corresponding private key.

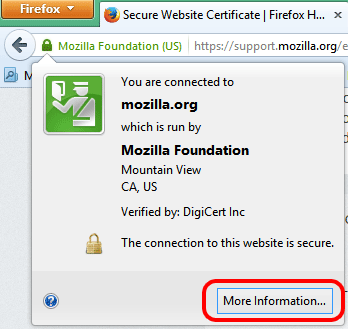
**Public Key Algorithm**: Algorithm used to create the Public Key.

**Fingerprint**: An abbreviated form of the Public Key.

**Fingerprint Algorithm**: Algorithm used to create the Fingerprint.

**View a Certificate**

When you have browsed to a website whose web address starts with https, there will be a lock icon at the beginning of the address bar. Single-click on the lock icon to get a pop-up that says who verified the certificate, then click on More Information….

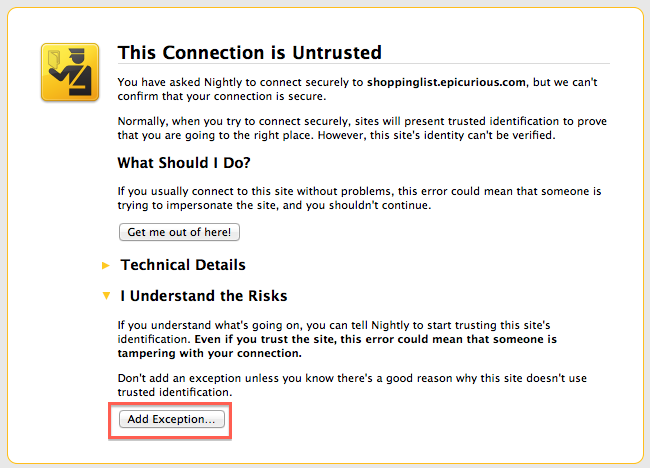


In that window, click on **Security**, then View Certificate.

When you browse to a website whose web address starts with https and there is a problem with the Secure Website Certificate, you will see the [This Connection Is Untrusted](https://support.mozilla.org/en-US/kb/connection-untrusted-error-message) alert page. Some common errors are described [here](https://support.mozilla.org/en-US/kb/connection-untrusted-error-message#w_technical-information).

To view the problematic certificate, follow these steps:

1. On the warning page, click **I Understand the Risks**.
2. Click Add Exception….



1. When the Add Security Exception dialog appears, click View…. The Certificate Viewer dialog will appear.

After you encounter an Untrusted Connection error, you may see a popup window asking you to report the error to Mozilla. Sharing the address and site identification (the Secure Website Certificate) for the site that was untrusted will help us identify and block malicious sites to keep you better protected.

## What is encryption and why are there different levels?

Encryption is a mathematical process of coding and decoding information. The number of bits (40-bit, 56-bit, 128-bit, 256-bit) tells you the size of the key. Like a longer password, a larger key has more possible combinations. In fact, 128-bit encryption is **one trillion times stronger** than 40-bit encryption. When an encrypted session is established, the strength is determined by the capability of the web browser, SSL certificate, web server, and client computer operating system.

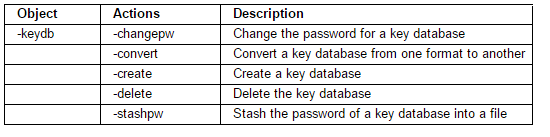
## How does SSL make my web site more trustworthy?

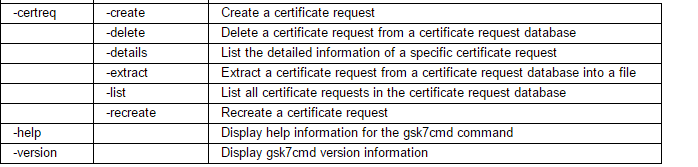
An SSL certificate contains verified information about the web site it secures to help users confirm that they are communicating with your web site. Extended Validation is the industry’s highest standard of verification and provides the most visible assurance to users: the **address bar turns green** in high-security browsers.

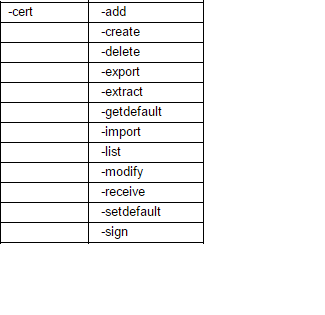
When you display the Thawte Trusted Site Seal, users can click the trust mark to view web site identification information, the third party (such as Thawte) that verified it, and the expiration date of the SSL certificate. In newer browsers, web site identification information may appear when users hover over the address bar. They can also click the closed padlock icon.

### GSK7CMD

By using the GSK7MD command we can perform the various operations related to certificate work. Below we are listing the available options for GSK7CMD command.







**Ikeymantool**

The key-management program, **iKeyman**, is provided with IBM® SDK Java™ Technology Edition. It is a user-friendly GUI for managing key files, which is implemented as an applet.

# An overview of the SSL or TLS handshake

The SSL or TLS handshake enables the SSL or TLS client and server to establish the secret keys with which they communicate.

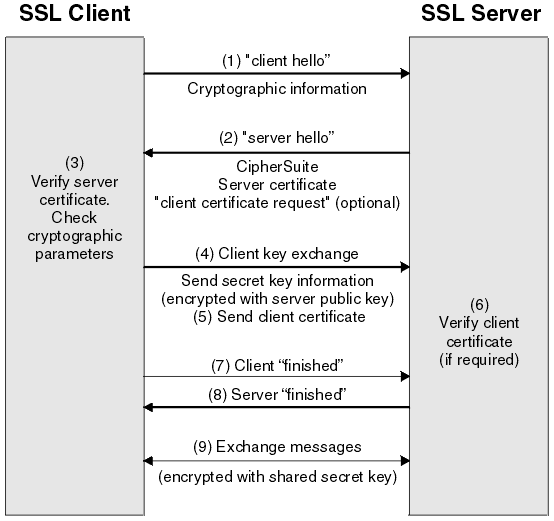
This section provides a summary of the steps that enable the SSL or TLS client and server to communicate with each other:

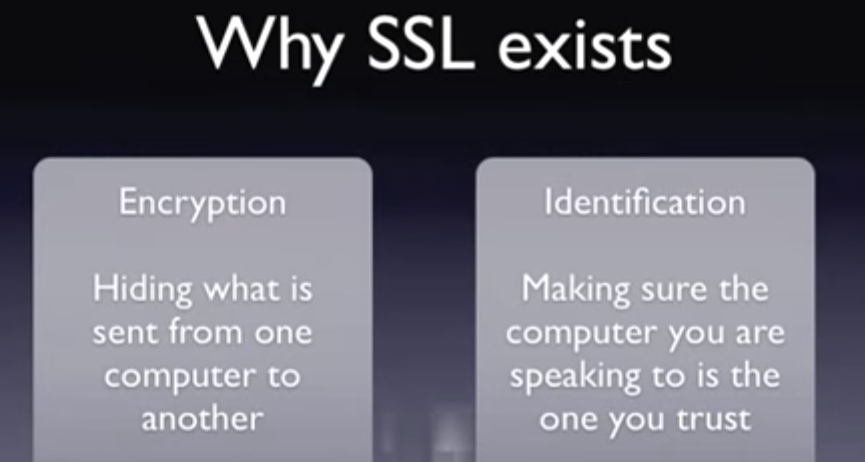
* Agree on the version of the protocol to use.
* Select cryptographic algorithms.
* Authenticate each other by exchanging and validating digital certificates.
* Use asymmetric encryption techniques to generate a shared secret key, which avoids the key distribution problem. SSL or TLS then uses the shared key for the symmetric encryption of messages, which is faster than asymmetric encryption.

For more information about cryptographic algorithms and digital certificates, refer to the related information.

This section does not attempt to provide full details of the messages exchanged during the SSL handshake. In overview, the steps involved in the SSL handshake are as follows:

*Figure 1. Overview of the SSL or TLS handshake*





**Why encryption**-Suppose, we are having communication mechanism, where two party is taking to each other, if here let’s say both the party is sharing the critical details, so those information can be hacked over the network, therefore for securing such issues we are having SSL, so only intended party can see/use the information.



