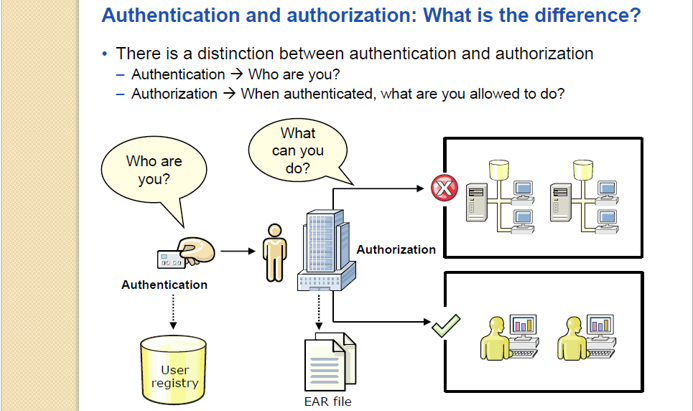
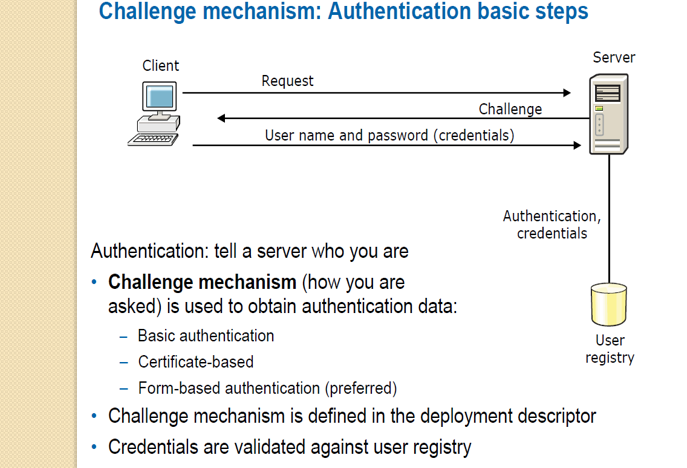
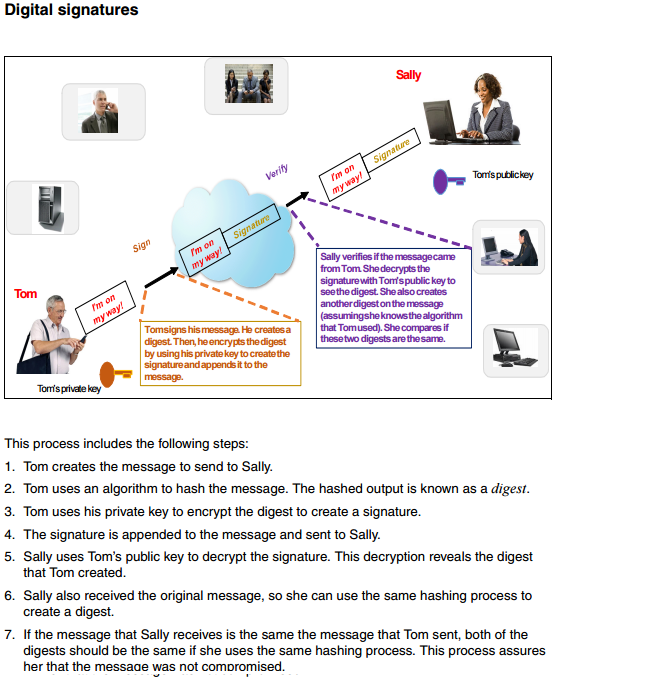
**WebSphere Security**



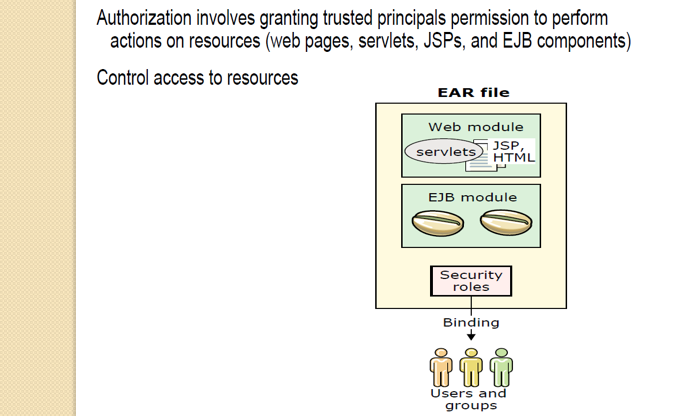




How does Sally get Tom’s public key? How does Sally know which algorithms to use to create the hash?

The answer to these questions is the digital certificate. This certificate is the container for the public key and the algorithm. It also indicates the owner of the certificate and other information.

**Application security**



Application is a peace of code that satisfies the end user request; let’s have an example of Facebook.

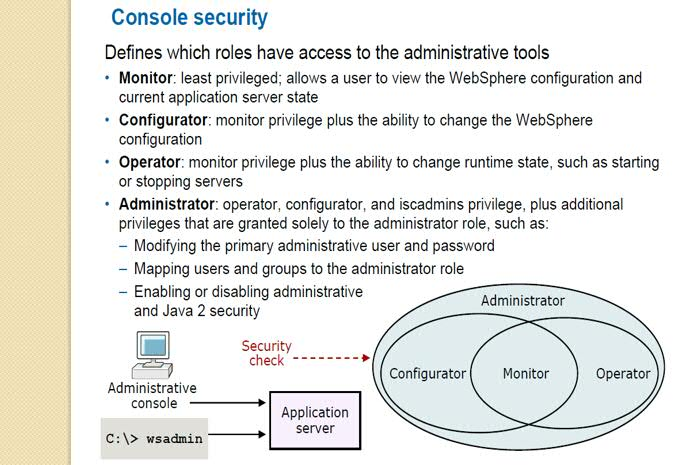
Facebook is an application, and only authorized person can use the application.

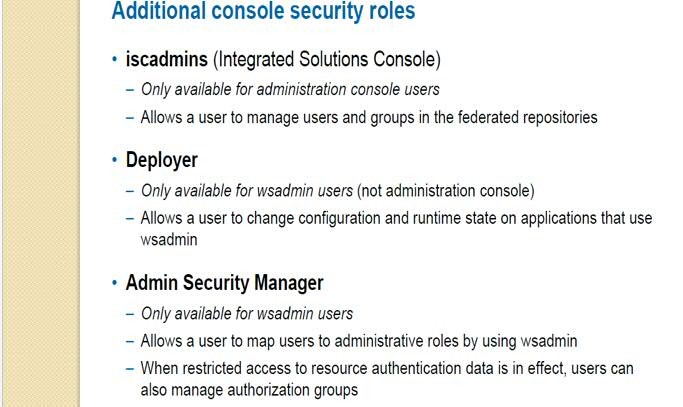


Here the Facebook is our application, which has been secured by userid /password.

So, only authorized user can access the application.

**Admin Security**





**J2EE security**

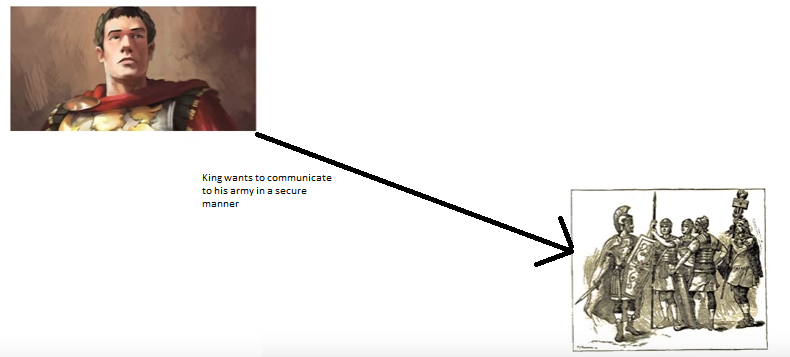
It is all about the certificate that we installed for making application secure.

**Cryptography** is a method of storing and transmitting data in a particular form so that only those for whom it is intended can read and process it.

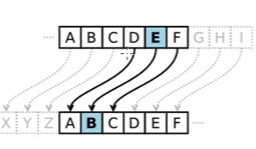
**Algorithm** is a process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer.

Here, king wants to send the secure message to his army with two conditions- 1) Message should be read/understand by his army only 2) Correct message needs to deliver.

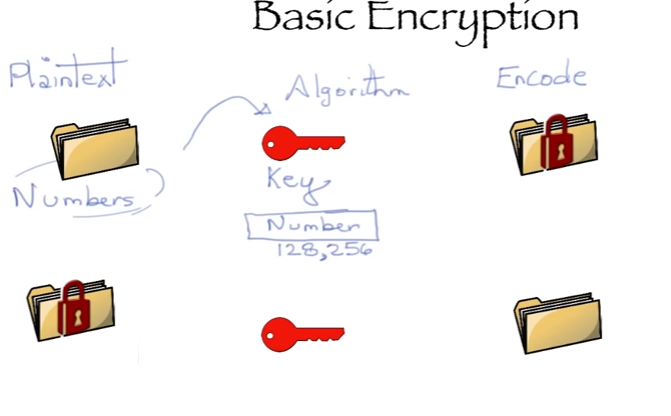
He he opted for secure communication



So he used encryption method- replacing the alphabets.

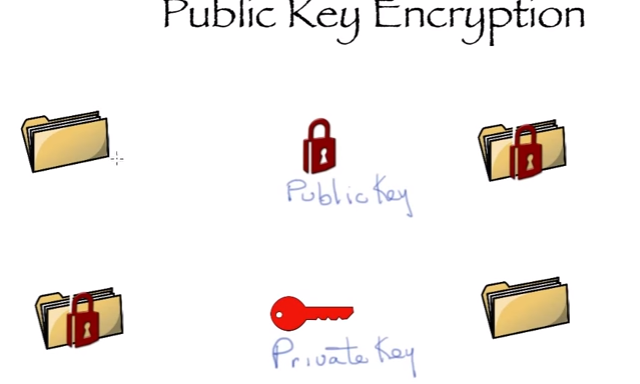


This encryption is taken care by the algorithms.

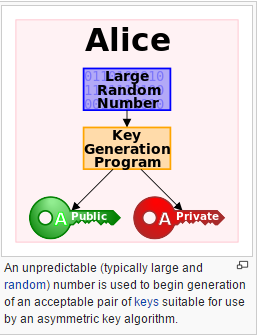


Symmetric encryption- when we are using the same key for encryption and decryption.

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.

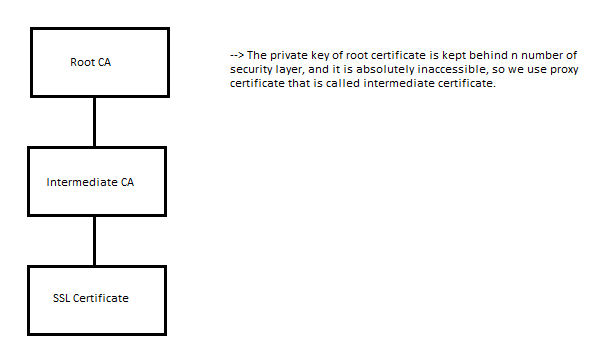
**What is an SSL Certificate?**

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.

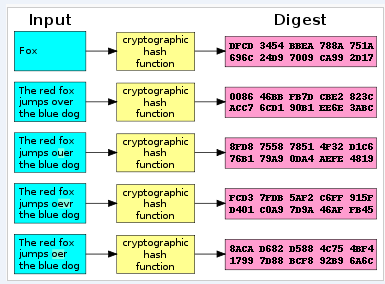
## Why does my website need an SSL Certificate?

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.

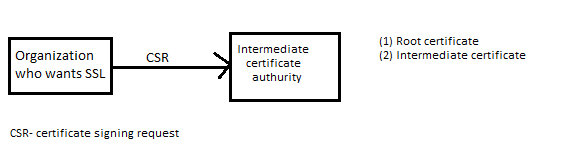
**Chain of trust**

**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).

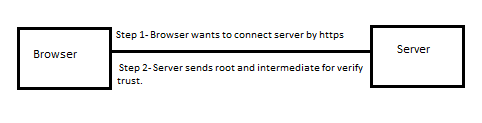


**What Happens when any organization wants security?**

****

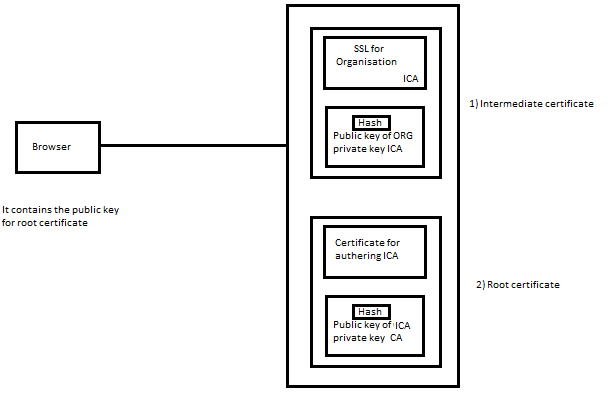
**What Happens Between the Web Browser and Server?**

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.

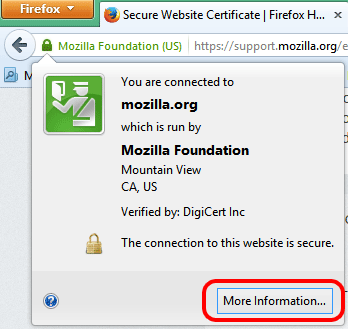
**Public Key Algorithm**: Algorithm used to create the Public Key.

**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….

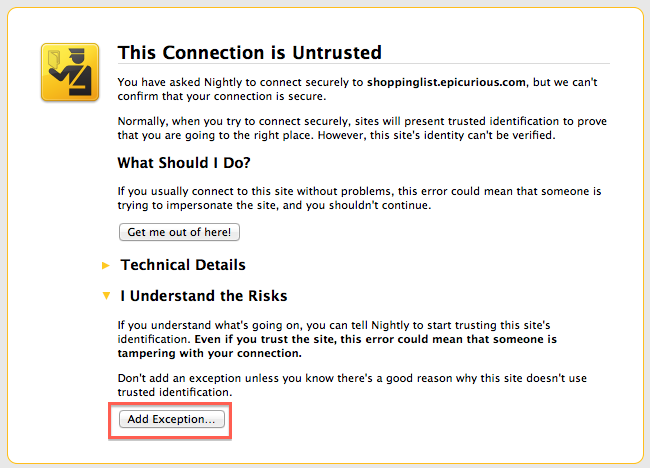


In that window, click on **Security**, then View Certificate.

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.

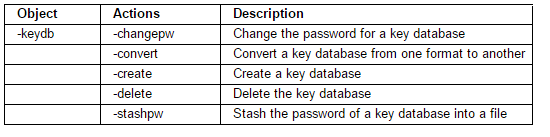
## How does SSL make my web site more trustworthy?

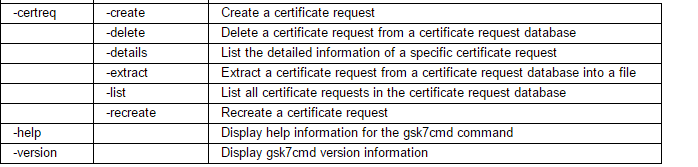
An SSL certificate contains verified information about the web site it secures to help users confirm that they are communicating with your web site. Extended Validation is the industry’s highest standard of verification and provides the most visible assurance to users: the **address bar turns green**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

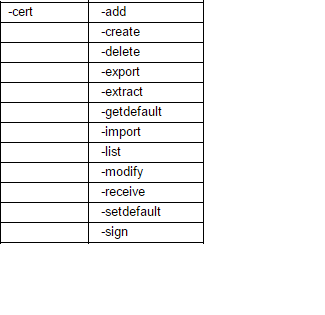
When you display the Thawte Trusted Site Seal, users can click the trust mark to view web site identification information, the third party (such as Thawte) that verified it, and the expiration date of the SSL certificate. In newer browsers, web site identification information may appear when users hover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

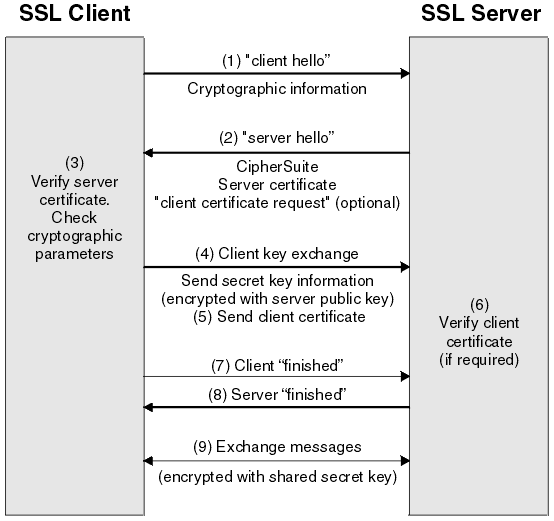
This section provides a summary of the steps that enable the SSL or TLS client and server to communicate with each other:

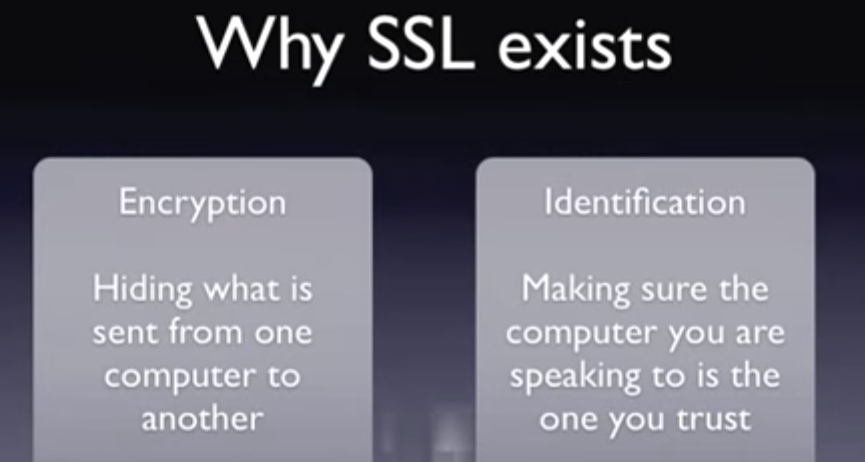
* Agree on the version of the protocol to use.
* Select cryptographic algorithms.
* Authenticate each other by exchanging and validating digital certificates.
* Use asymmetric encryption techniques to generate a shared secret key, which avoids the key distribution problem. SSL or TLS then uses the shared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

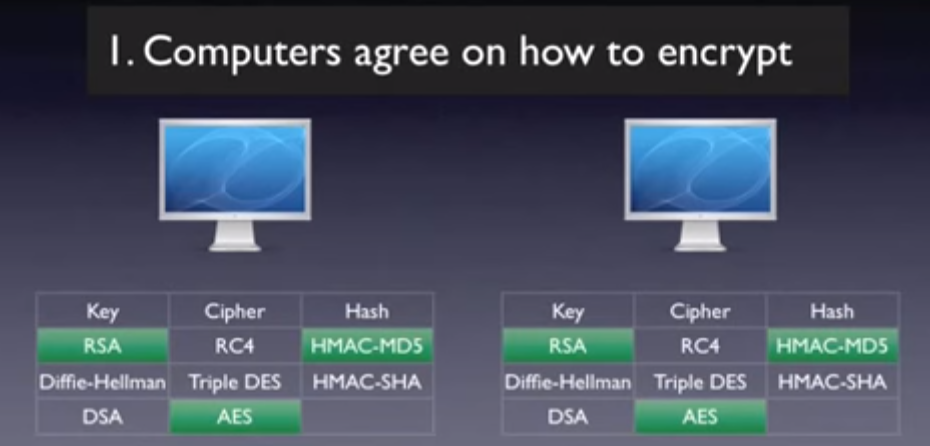
*Figure 1. Overview of the SSL or TLS handshake*

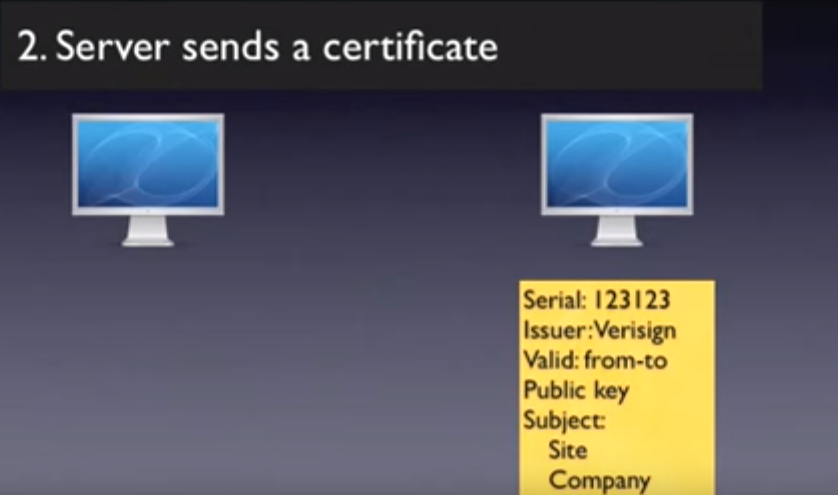


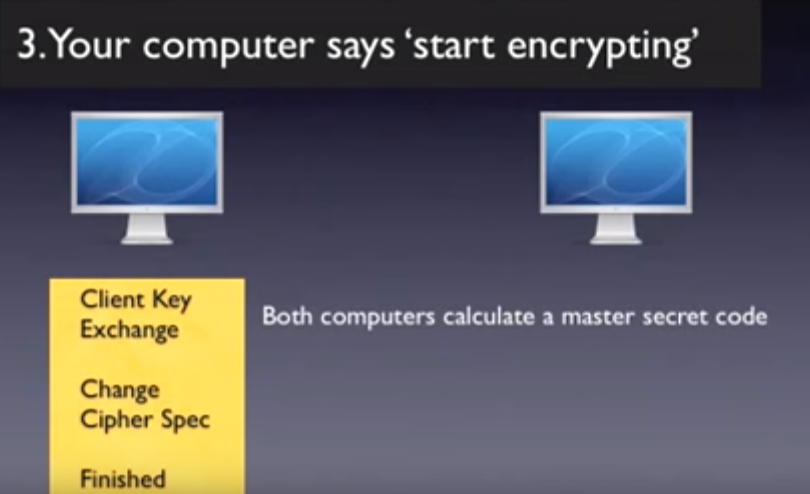


**Why encryption**-Suppose, we are having communication mechanism, where two party is taking to each other, if here let’s say both the party is sharing the critical details, so those information can be hacked over the network, therefore for securing such issues we are having SSL, so only intended party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



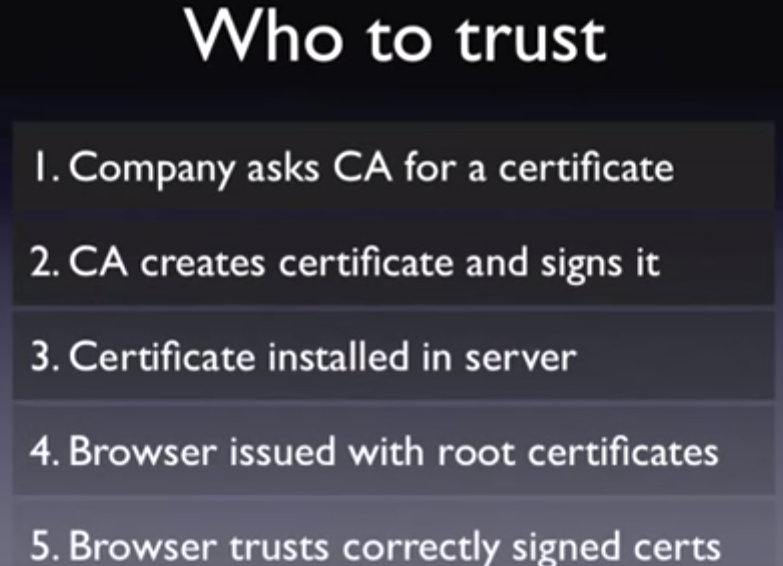












**Certificate Prank**





In the today’s life in order to avoid such situations we have certificate authorities, that issue the certificate and provide the security. We are having two types of certificate authority- 1) Root certificate authority 2) Intermediate certificate authority

Here intermediate certificate authority issues the certificate on the behalf of root certificate authority.

Digital signature- Private/public key is called the digital signature.

We represent private key by K- and public key by K+.

|  |  |
| --- | --- |
| Term | Description |
| Trust store | In trust store we are having list of party with whom we can communicate |
| Key store | In Keystore we are having Key(private and public key) |

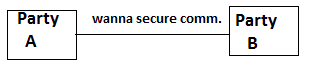
**SSL certificate(Secure socket layer)**

If you want to have the secure connection between two parties then you need to enable the SSL between them.

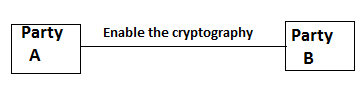
For enabling the SSL certificate we are having the two ways- 1) GUI based ikeyman tool, 2) Command line utility gsk7cmd command.

**Basics of SSL work**

Suppose, we are having two parties who want to communicate with secure manner then we will be implementing the SSL between them.



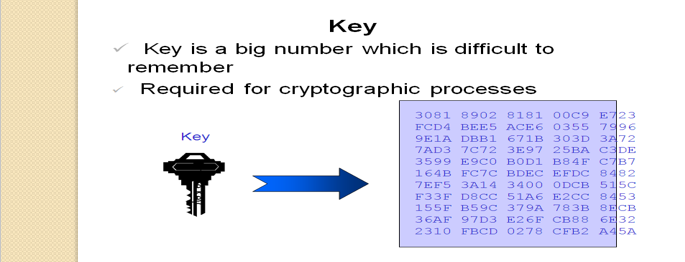
**Answer is**- Enable the Cryptography between them.

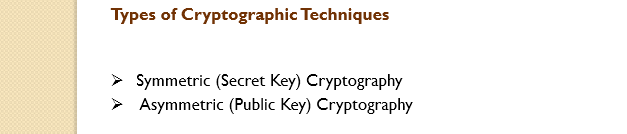


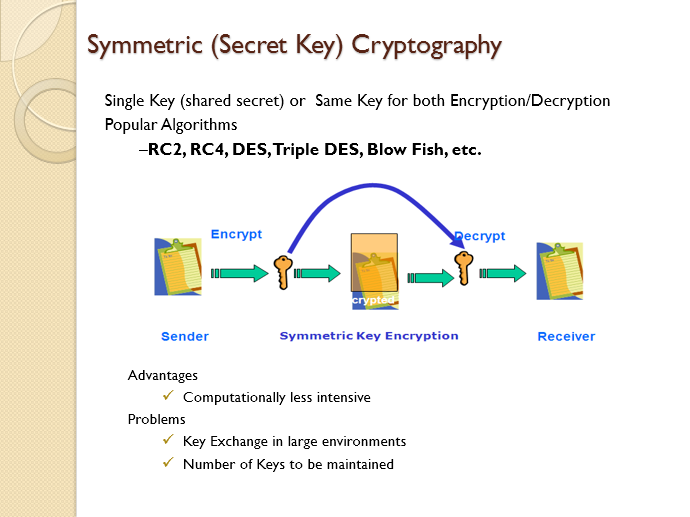
Once you will enable the SSL then, Party A and Party B can communicate in secure manner. **When party A is sending message to party B then message will be encrypted by the public key at party A level and decrypted at party B level by private key.**

**Cryptography**

Cryptography is a method of storing and transmitting data in a particular form so that only those for whom it is intended can read and process it.







What is keystore-

A keystore contains private keys, and the certificates with their corresponding public keys.

What is Trust store-

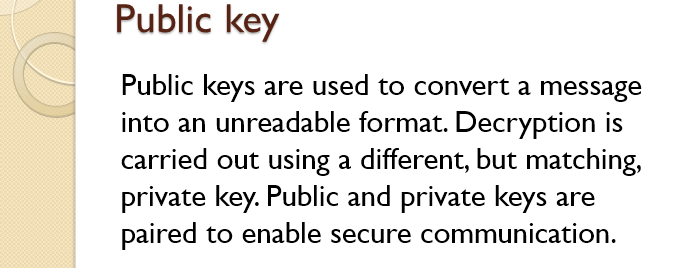
A trust store contains certificates from other parties that you expect to communicate with, or from Certificate Authorities that you trust to identify other parties.

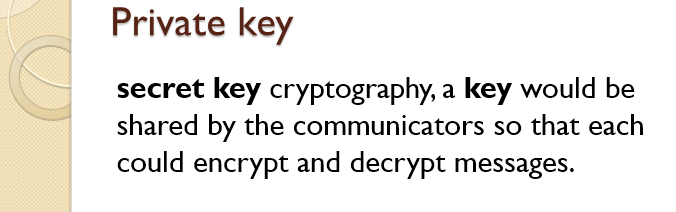
**Two Key concept**

In the two key concept, we will be having two key between the parties, namely public key and private key.

By the public key, we will be encrypting the message and with private key we will be decrypt the message. These two keys is commonly known as key pair.







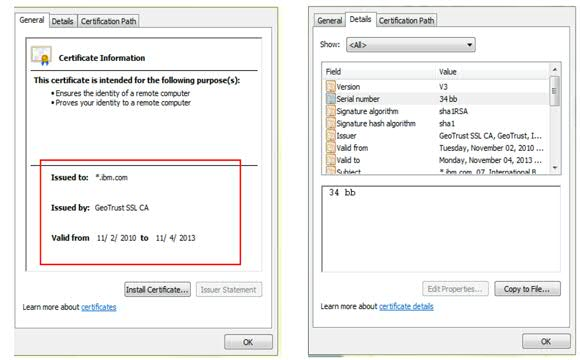


**SSL certificate**

The SSL certificate is the most trustable way for the communication between two parties.



**What information it contain?**



**How SSL certificate work**

 # Suppose, party A wants to talk to party B, then party A will be sending the request by web browser.

# Then party B will verify and generate the form based authentication (SSL communication).

# Party A will the form and encrypt the message with public key.

# Party B will verify the request in trust store, then decrypt the message by private key and SSL session will be established.

**Example of certificate name**

Certificate\_name.DER

**Managing SSL certificate by gsk7cmd command**

Command: gsk7cmd Purpose: gsk7cmd is a command line tool for certificate management

**Prerequisite: set JAVA\_HOME variable.**

Example: export JAVA\_HOME=/usr/IBM/WebSphere/AppServer/java (this depends on your environment)

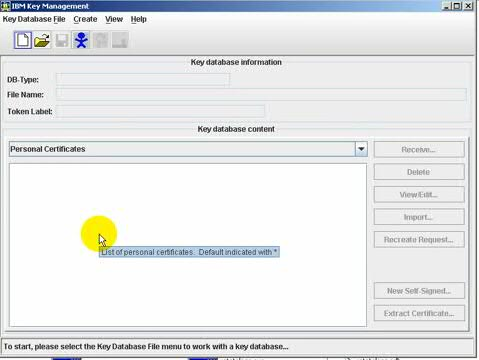
# gsk7cmd -help  
  
Object Action Description  
------ ------ -----------  
**-keydb**   
            -changepw Change the password for a key database  
            -convert Convert the format of a key database  
            -create Create a key database  
            -delete Delete a key database  
            -expiry Display password expiry  
            -list Currently supported types of key database.  
            -stashpw Stash the password of a key database into a file  
  
**-cert**   
             -add Add a CA Certificate  
             -create Create a self-signed certificate  
             -delete Delete a certificate  
             -details Show the details of a specific certificate  
             -extract Extract a certificate from a key database  
             -getdefault Show the default personal certificate  
             -import Import a certificate from a key database or a PKCS12 file  
             -list List certificates in a key database  
             -listsigners List signer certificates delivered with ikeyman  
            -modify Modify a certificate

**-certreq**  
              -create Create a certificate request  
              -delete Delete a certificate request from a certificate request database  
              -details Show the details of a specific certificate request  
              -extract Extract a certificate from a certificate request database  
              -list List all certificate requests in a certificate request database  
              -recreate Recreate a certificate request  
  
**-seckey**  
  
             -create Create a secret key  
             -delete Delete a secret key  
             -details Show the details of a specific secret key  
             -export Export secret keys to a file  
             -import Import secret keys from a file  
             -list List all secret keys in a key database  
             -rename Rename a secret key  
  
**-version**   Display iKeyman version information  
  
**-help**        Display this help text

Creating the certificate with ikeyman(GUI way to manage certificate)

------------------------------------------------------------------------------------

Step 1-> Open the Ikeymantool and get the certificate.



We will create one more folder in the IHS root called IHS

Go to keydata base file, select new then create the below file

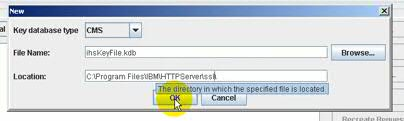
The Ikeyman tool

----------------------

The Ikeyman tool is used to create the certificate.

How to create the certificate

---------------------------------



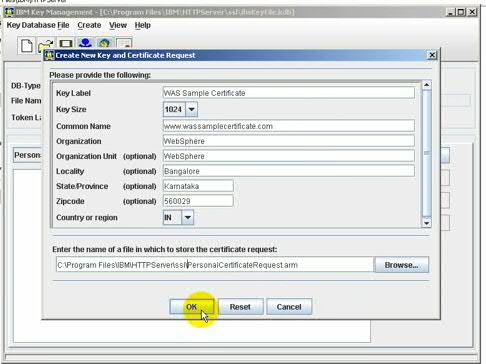
for time being select the password as 123456789.



set certification expire date is 90 days and click on stash the password, it will encript the password.



Now we have to create the certificate, now select the personal certificate and click on new then you will get the below form

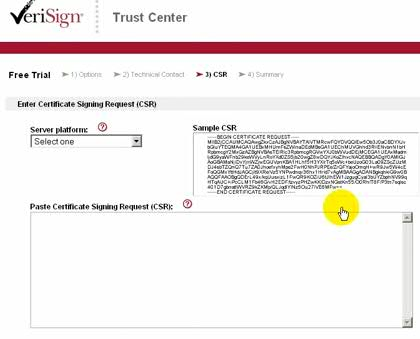


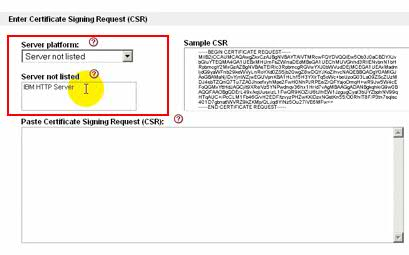
Now click on OK.

Now close the ikeyman and go to the website which provides the free certificate(exp verision)



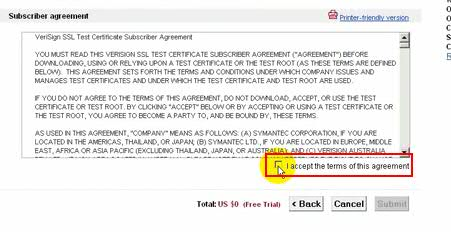
Pls provide the details inorder to get the certificate, whatever emailid you will provide in the same id you will get certificate in the mail, once you click on continue then you will get the below page

Now defining the server



As we have created the personal certificate request in the SSL, go to that file open that .arm file, copy that paste it here and click on continue.

Now accept the terms and conditions of verisign and click on submit.



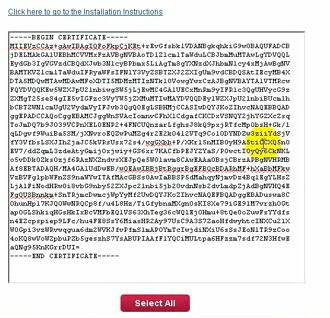
Then you will get an email stating that your order has been complsted. And you will get certificate in your email id

then go to email and download the certificate, in the email

We will get the three types of certificate-

1. Root certificate.
2. Intermediate certificate.

The certificate looks like below-



click on the select all the copy, then now go to SSL folder and create a new file name Rootcertificate.arm, name can be anything but extension should be .arm.

Step 1-> root certificate.

Step 2-> intermediate certificate.

Step 3-> we will get the code to download the SSL in the mail, copy that code and put into the SSL folder.

Now go to IHS bin directory and execute the ikeymantool.sh

from keydatabase side select open



click on ok and give the password.

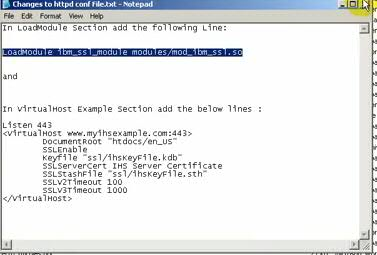
Now click on signed cerificate and add root and intermediate certificate, while enering the

certificate details pls give label name for certificate as root certificate or intermediate certificate.



Now select the personal certificate and add the final certificate here.

Now, our next step is configuring virtual host, open the httpd conf file and add the deails.







Then provide the key file location.

Save the containt.

Now go to etc folder.

Path for ETC folder is ->[C:/windows/system32/drivers/etc/host](file:///C:\Users\juhi3\Desktop\Ganesh%20Kripa%20Devops%20Engineer\Ganesh_Devops\WC_devops\windows\system32\drivers\etc\host)

open the host file-> assign ip to the domain



Now recycle the webserver and generate the plugin

we can generate the plugin from the bin directory of DMGR.

Incase you found the plugin details is not there then add the plugin details in the httpdconf file.



**IBM WebSphere Application Server (WAS) - SSL HANDSHAKE**

SSL Handshake means after the installation of the certificate we will be doing the basic connectivity test.

**IBM SSL handshake Failure**

## Resolving the problem-A Java application that makes a secure connection to a Web site was failing. There were no obvious messages in the System.out log.

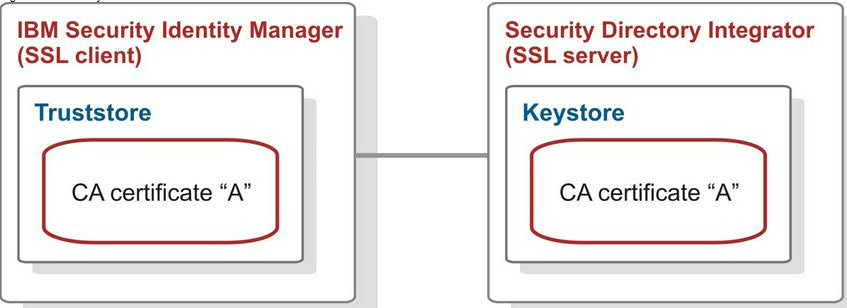
WAS security tracing was enabled by:  
1. Going to the WebSphere Administrative Console.  
2. In the menu on the right, select **Troubleshooting**, then select **Logs and Tracing.**  
3. Enable \*all tracing for the following:  
**com.ibm.websphere.security.\***  
**com.ibm.ws.ssl.** We then restarted the server, re-created the issue, and received the following exception in the System.out log:

**Why we need the SSL certificate**

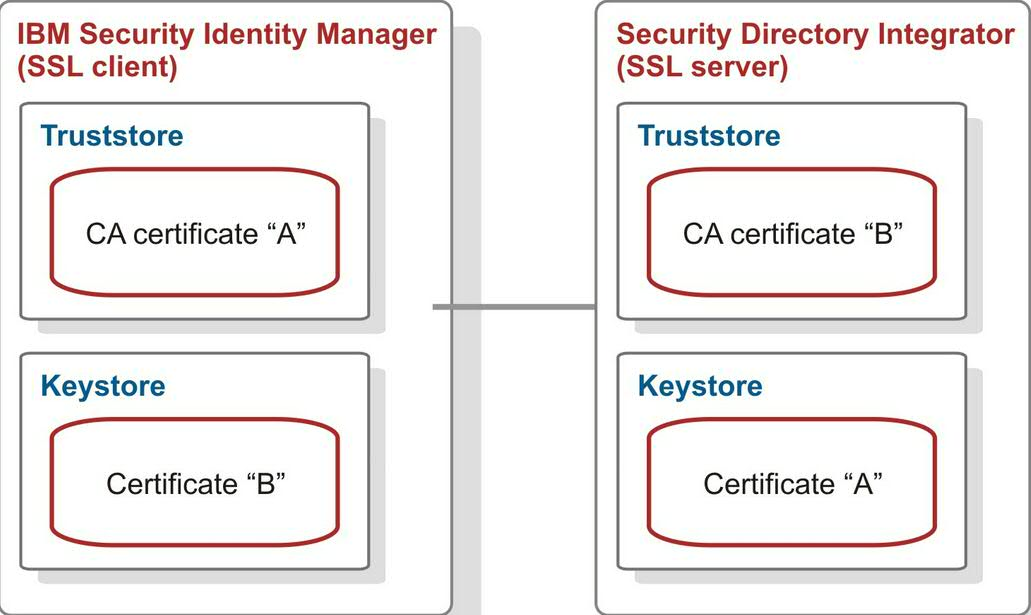
SSL certificate provides the secure communication between parties, there are two ways of SSL communication-

**One way SSL communication**

In one way SSL communication, when client want to communicate to server then server share his identity with Client.



Two way SSL communication, server and client, both share the identity with each other.



**Keystore** - In the key store, we are having the public key and private key.

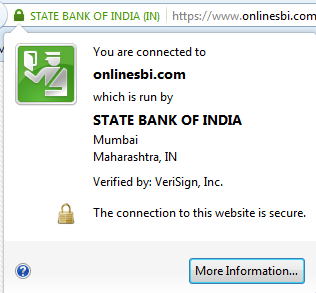
**Trust store** – In the trust store the certificate from the parties, whom we want to communicate.

Example- Flipkart is having many parties, so all the other parties whom Flipkart wants to communicate is having there certificate at Flipkart trust store.

**How to check a website is secured or not and how?**

Once you open the site, you will be getting the lock symbol





**Type of certificate**

We are having two types of certificate –

1. Personal certificate
2. Digital certificate

**Personal certificate**

We can create the personal certificate by own, it is also known as self-signed certificate.

**Digital certificate**

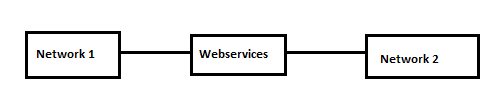
Digital certificate is provided by the certificate authorities.

Certificate authorizes study about the environment and provide the certificate.

**Certificate authorities**

Certificate authority is a third party organization, who studies about the environment and provides the appropriate certificate.

**Webservices**- Webservices provides interface between two network.



**LTPA Token**

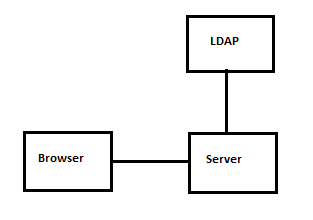
Lightweight Third-Party Authentication (LTPA) is a type of authentication mechanism in WebSphere® Application Server security that defines a particular token format. The purpose of the LTPA token authentication is to flow the LTPA token from the first Web service, which authenticated the originating client, to the downstream Web service. You can configure the server to validate LTPA token authentication.

**SSO**

Single sign-on (SSO) is a [session](http://searchsoa.techtarget.com/definition/session) and user [authentication](http://searchsecurity.techtarget.com/definition/authentication) service that permits a user to use one set of [login](http://searchsecurity.techtarget.com/definition/logon) credentials (e.g., name and password) to access multiple applications. The service authenticates the end user for all the applications the user has been given rights to and eliminates further prompts when the user switches applications during the same session. On the back end, SSO is helpful for logging user activities as well as monitoring user accounts.

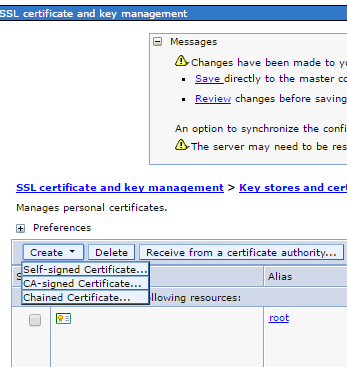
Example- Once we login to gmail, by default we login to all the application of google like google map, google Images… many more.For implementing the SSO, we need to create the LTPA token from one application and we need to import that into another application.

**LDAP** (Lightweight Directory Access Protocol) is a tool for authorization purpose; we are having LDAP servers that saves the user related information



**Enabling Global security**

1. Create the communication between node and dmgr.
2. Enable the security
3. Assign the roles
4. Restart the processes



In the chained certificate we are having the multiple certificate.

Root-> Chetan-> Mahesh-> Krishna->Umesh(Umesh want certificate and CA can’t authorized him so he used third person as a member to get him introduce)

**SSL configuration**

The Secure Sockets Layer (SSL) protocol provides secure communications between remote server processes or endpoints. SSL security can be used for establishing communications inbound to and outbound from an endpoint. To establish secure communications, a certificate and an SSL configuration must be specified for the endpoint.

In previous versions of this product, it was necessary to manually configure each endpoint for Secure Sockets Layer (SSL). In this version, you can define a single configuration for the entire application-serving environment. This capability enables you to centrally manage secure communications. In addition, trust zones can be established in multiple node environments by overriding the default, cell-level SSL configuration.

If you have migrated a secured environment to this version using the migration utilities, the old Secure Sockets Layer (SSL) configurations are restored for the various endpoints. However, it is necessary for you to re-configure SSL to take advantage of the centralized management capability.

