

UD2. Configuración segura de servidores web



Despliegue de Aplicaciones Web
2º DAW

ÍNDICE

- CONFIGURACIÓN DESCENTRALIZADA
- MODELOS DE AUTENTIFICACIÓN: BÁSICA Y DIGEST
- PROTOCOLO HTTPS
- CERTIFICADOS Y AUTORIDADES DE CERTIFICACIÓN
- CONFIGURACIÓN **SSL/TLS** DE SERVIDOR:
CERTIFICADOS DE SERVIDOR

1. CONFIGURACIÓN DESCENTRALIZADA

- No siempre el **desarrollador** y/o **clientes** que comparten un mismo servidor pueden tener acceso para administrarlo.
 - Servicio de **hosting**.
 - Departamento **de sistemas y desarrollo**
- Se necesita de algún mecanismo para que cada **cliente** pueda gestionar su propia **configuración** sin que ello implique la manipulación del servidor **http**



1. CONFIGURACIÓN DESCENTRALIZADA

- El fichero **.htaccess** nos permite realizar configuraciones distribuidas del servidor web, en lugar de **centralizadas**, en un solo fichero de configuración.
 - Permite **modificar la configuración** principal según el **directorio** donde se sitúe el fichero **.htaccess**.
- Todas las directivas de configuración se aplican al **directorio** y **subdirectorios** donde está situado el fichero **.htaccess**.



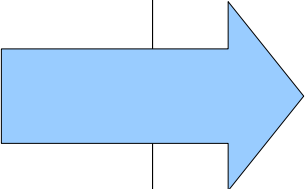
1.1 SOBRESCRITURA DE CONFIGURACIÓN. FICHERO .HTACCESS

- Este tipo de configuración distribuida solo se ha de realizar cuando se quiere compartir el servidor web y no se puede dar permiso a todos los administradores al fichero de configuración principal.
- Es necesario tener en cuenta que la utilización de ficheros `.htaccess` **disminuye el rendimiento** del servidor y, siempre que sea possible, se ha de intentar evitar.
- Para poder evitar esto, se realizan las configuraciones en el **fichero de configuración** de cada **host virtual** utilizando la directiva `<Directory>`.

1.1 SOBRESCRITURA DE CONFIGURACIÓN. FICHERO .HTACCESS

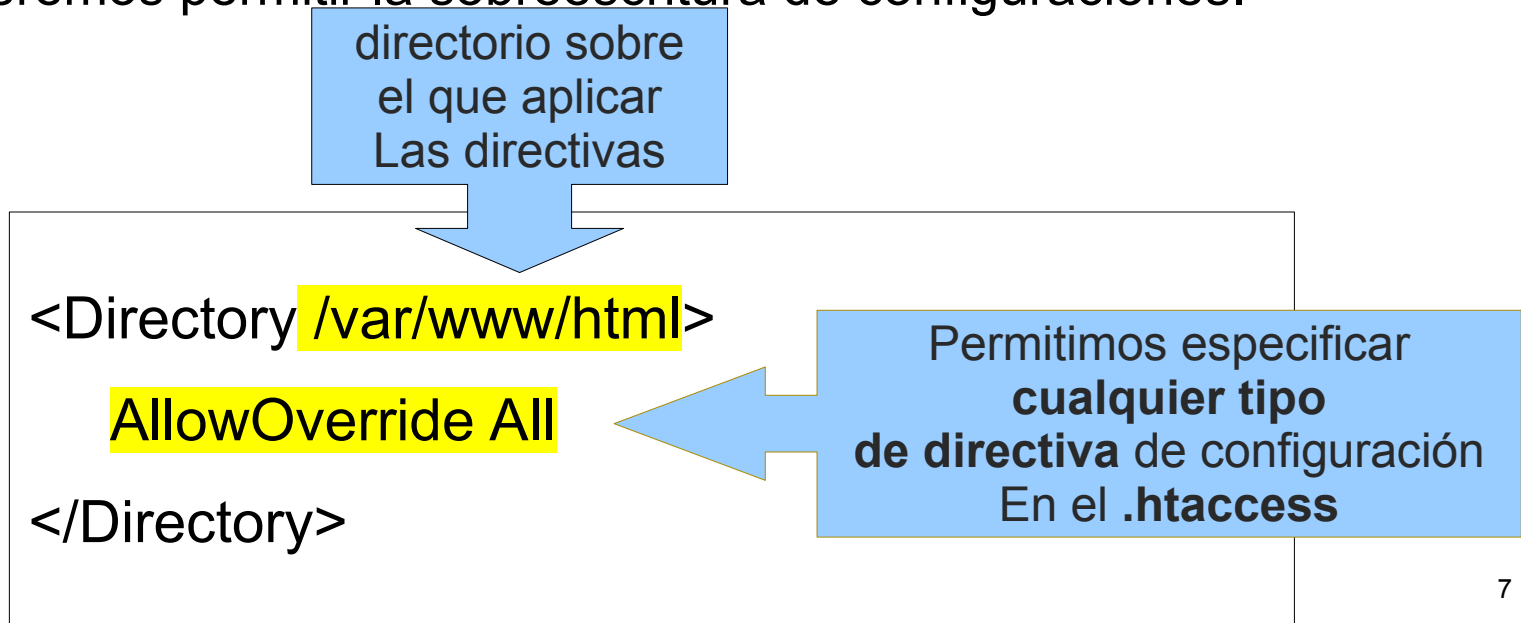
Configuración en virtualHost vs .htacceess

<i>/etc/sites_available/example.conf</i>	<i>/var/www/html/.htaccess</i>
<pre> <VirtualHost *:80> ServerName www.example.com ServerAdmin webmaster@localhost DocumentRoot /var/www/html ... <Directory /var/www/html> Directiva 1; Directiva 2; Directiva 3; </Directory> </VirtualHost> </pre>	<pre> Directiva 1; Directiva 2; Directiva 3; </pre>



1.1 SOBRESCRITURA DE CONFIGURACIÓN. FICHERO .HTACCESS

- Para poder aplicar directivas en el fichero `.htaccess` se ha de permitir en la configuración del vhost
- Se realizará con la directiva `AllowOverride` dentro de un tag `<Directory>` que haga referencia al directorio en el que queremos permitir la sobreescritura de configuraciones.



1.1 SOBREESCRITURA DE CONFIGURACIÓN. FICHERO .HTACCESS

- Las modalidades u opciones de la directiva AllowOverride son:
 - **All:** permite utilizar cualquier directiva de configuración.
 - **None:** permite utilizar ninguna directiva de configuración.
 - **AuthConfig:** permite utilizar directivas de autorización.
 - **FileInfo:** permite utilizar directivas para controlar los tipos de documentos. (Error Document, Rewrite Rules,...)

1.1 SOBREESCRITURA DE CONFIGURACIÓN. FICHERO .HTACCESS

- **Indexes:** permite utilizar directivas relacionadas con el listado de directorios.
- **Limit:** permite utilizar directivas relacionadas con las listas de control de acceso al servidor.
- **Options:** permite especificar directivas relacionadas con características de los directorios.

1.1 SOBRESERITURA DE CONFIGURACIÓN. FICHERO .HTACCESS

- Podemos encontrar más información sobre las directivas en la documentación oficial:
- ***Ejemplo:** Si queremos permitir la utilización de directivas para las páginas de error mediante `.htaccess` en el directorio `"/usr/local/apache2/www/aplicacioDAW"` añadimos:*

```
<VirtualHost *:80>  
  <Directory /usr/local/apache2/www/aplicacioDAW>  
    AllowOverride FileInfo  
  </Directory>  
</VirtualHost>
```

2. MODELOS DE AUTENTICACIÓN: BASIC Y DIGEST

- Los servidores web proporcionan mecanismos de autenticación:
 - **La autenticación:** verifica que alguien es quien dice ser y se basa en un nombre de usuario y una contraseña.
- Los usuarios y sus contraseñas se guardan en un repositorio o proveedor de autenticación, por ejemplo un fichero o una base de datos.

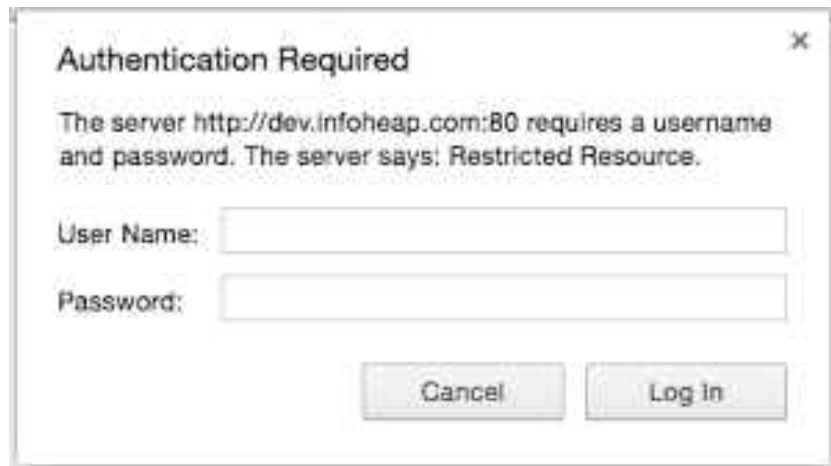


2. MODELOS DE AUTENTICACIÓN: BASIC Y DIGEST

- El **servidor web Apache** utiliza diferentes módulos para implementar estos mecanismos de seguridad. Los podemos ver en la documentación oficial:

<https://httpd.apache.org/docs/current/es/howto/auth.html>

- Estos sistemas nos pueden ser útiles cuando nuestro sitio web tiene información sensible o dirigida solo a un pequeño grupo de personas.



2. MODELOS DE AUTENTICACIÓN: BASIC Y DIGEST

- **Modelo basic:** no utiliza ningún tipo de mecanismo criptográfico para asegurar los datos, que viajan en abierto dentro de las peticiones HTTP.
- **Modelo Digest:** utiliza criptografía simétrica para cifrar los datos y asegurar la confidencialidad. Pero no es del todo seguro, ya que hay un intercambio previo de las claves simétricas, que se transmiten en abierto por la red. Por tanto, cualquiera puede interceptarlas para poder descifrar posteriormente los datos.

Nota: Criptografía simétrica. La simetría está en que la clave de cifrado es la misma que en proceso inverso, el descifrado.

2. MODELOS DE AUTENTICACIÓN: BASIC Y DIGEST

- Modelo de autenticación Basic:
 - Primero: fichero con contraseñas.
 - Este fichero se ha de crear en un sitio que no sea accesible desde la web. Ejemplo:
 - 4 *`/var/www/html/app1`*
 - 5 *`/var/www/passwd/app/`*
 - Para crear el fichero de contraseñas se hará uso de la utilidad que viene con apache2 llamada htpasswd, que se encuentra en el directorio `/usr/bin/`, de la siguiente forma:

```
$ htpasswd -c /var/www/passwd/passwords nombreUsuari
```

Nota: Para añadir sucesivas contraseñas lo haremos sin el parámetro `-c`.

2. MODELOS DE AUTENTICACIÓN: BASIC Y DIGEST

Modelo de autenticación Basic:

- El siguiente paso es configurar el servidor para que solicite una contraseña, además de especificar qué usuarios tienen acceso. Esto se realiza mediante el fichero **.htaccess**

.htaccess

```
AuthType Basic
```

```
AuthName "Restricted Files"
```

```
AuthUserFile /usr/local/apache2/passwd/passwords
```

```
Require user nomUsuari
```

2. MODELOS DE AUTENTICACIÓN: BASIC Y DIGEST

Directivas

- `AuthType`: Selecciona el método que se utiliza para autenticar al usuario. El más común es Basic (implementado en el módulo `mod_auth_basic`)
- `AuthName`: Cumple dos funciones importantes:
 - Presenta esta información al usuario como parte del cuadro de diálogo para introducir las credenciales.
 - Establecer un **dominio** y poder determinar qué contraseña enviar para cada **zona restringida**.

2. MODELOS DE AUTENTICACIÓN: BASIC Y DIGEST

Directivas

- **AuthUserFile** establece la ruta al fichero de contraseñas que acabamos de crear con htpasswd. Si tiene un gran número de usuarios, sería bastante lento buscar en el fichero de texto plano. Apache dispone de diferentes "Auth Providers" para almacenar la información del usuario en ficheros de bases de datos.

- Proveedor de Autenticación

- mod_authn_anon
- mod_authn_dbd
- mod_authn_dbm
- mod_authn_file
- mod_authnz_ldap
- mod_authn_socache

2. MODELOS DE AUTENTICACIÓN: BASIC Y DIGEST

Directivas

- **Require** proporciona la parte de autorización del proceso, estableciendo el usuario al que se le permite acceder a esta parte del servidor.
 - 4 **Require user nombre_usuario** : solo el nombre de usuarios que pueden acceder al recurso.
 - 5 **Require group nombre_grupo** : solo los usuarios que pertenecen al grupo pueden acceder al recurso.
 - 6 **Require valid-user** : todos los usuarios válidos (los que están en el fichero de contraseñas) pueden acceder al recurso.

2. MODELOS DE AUTENTICACIÓN: BASIC Y DIGEST

Directivas

- También podemos limitar el acceso a un fichero determinado del directorio, por ejemplo, si dentro de `/usr/local/apache/apache2/www/secret` tenemos el fichero `notes.html` lo limitamos mediante:

```
<VirtualHost *:80>  
    <Files notes.html>  
        Require user nomUsuari  
    </Files>  
</VirtualHost>
```

2. MODELOS DE AUTENTICACIÓN: BASIC Y DIGEST

- Modelo de autenticación Digest:
 - Primero es necesario activar el módulo **mod_auth_digest**.
 - A continuación se crea el fichero de contraseñas con la instrucción:

***htdigest -c /usr/local/apache2/passwd/pass
nombreDominio nombreUsuario***

2. MODELOS DE AUTENTICACIÓN: BASIC Y DIGEST

Modelo de autenticación Digest:

- El siguiente paso es configurar el servidor para que solicite una contraseña, y a qué usuarios se les permite el acceso.
- Esto se llevará a cabo mediante un fichero **.htaccess**
- Aplicamos las mismas directivas que con basic.

.htaccess

```
AuthType Digest
```

```
AuthName "Restricted Files"
```

```
AuthUserFile /usr/local/apache2/passwd/passwords
```

```
Require user nombreUsuario
```

2. MODELOS DE AUTENTICACIÓN: BASIC Y DIGEST

"Deberemos tener en cuenta que para que la configuración aplicada sea segura, se debería combinar con un certificado ssl, de forma que las credenciales y/o la clave de cifrado no viajen como texto plano."



3. PROTOCOLO HTTPS.

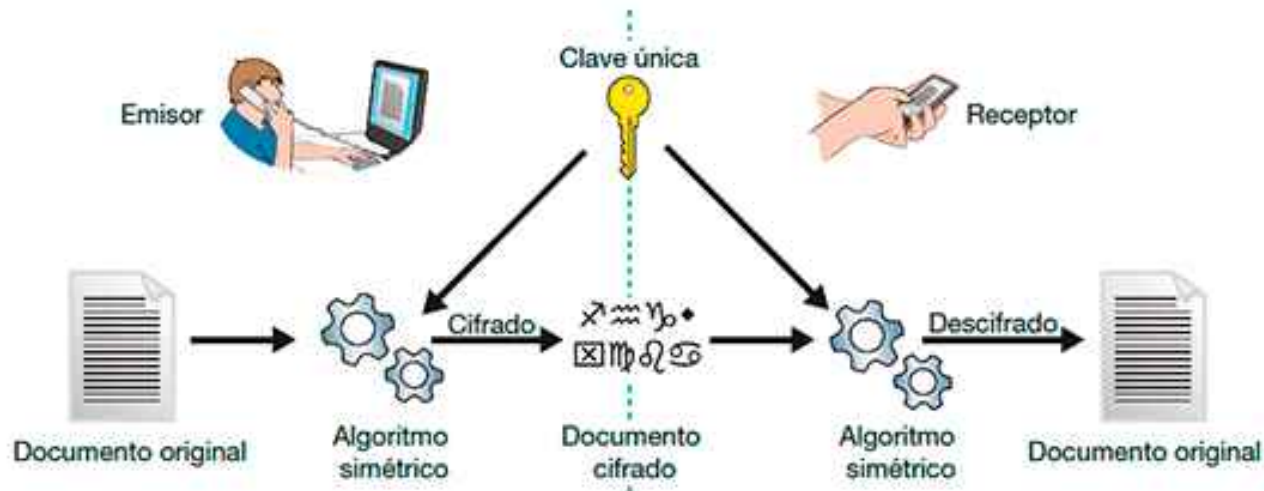
CONCEPTOS PREVIOS

¿Qué es la Criptografía?

- Técnica utilizada para convertir un texto claro en otro igual al anterior, pero que solo sea legible por personas autorizadas.
- **SSL** utiliza **diversos algoritmos** de encriptación y autenticación.
 - Para **establecer la conexión** con la máquina remota utiliza algoritmos de encriptación asimétrica.
 - Para la **transferencia de datos** utiliza algoritmos de encriptación simétrica, que son más rápidos.

3.1 CRIPTOGRAFÍA SIMÉTRICA

- Los algoritmos de criptografía simétrica son los que utilizan la misma clave tanto para el proceso de cifrado como el de descifrado.
- Los más utilizados: **DONES, 3DES, AES, IDEA y Blowfish**



Problema:
Intercambio
de claves

3.1 CRIPTOGRAFÍA ASIMÉTRICA

- Utiliza 2 claves matemáticamente relacionadas, de forma que se cifra con una (la clave pública) y se descifra con la segunda (clave privada).
- Algunos algoritmos representativos son: RSA, i *DSA



3.2 ¿QUÉ ES EL PROTOCOLO HTTPS?

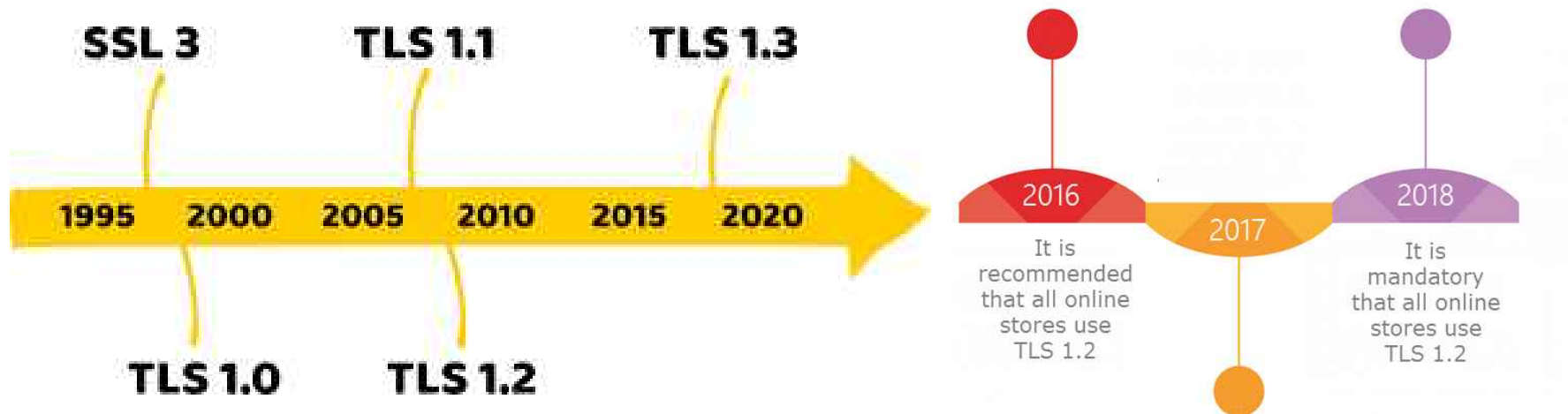
- ¿Qué es?
 - El protocolo **HTTPS** se basa en el **protocolo HTTP** y añade cifrado **SSL/TLS** para asegurar las conexiones entre emisor y receptor.
 - **HTTPS = HTTP + SSL/TLS**
 - Utiliza por defecto **el puerto 443**



3.2 ¿QUÉ ES EL PROTOCOLO HTTPS?

- **SSL/TLS**

- Protocolo seguro que pertenece a la capa de transporte. A lo largo del tiempo se han lanzado diferentes versiones, la mayoría de las cuales son **vulnerables**



3.3 CARACTERÍSTICAS

- El **protocolo SSL/TLS** proporciona las funcionalidades de confidencialidad, integridad y autenticación al protocolo de nivel superior (HTTP), utilizando mecanismos de criptografía tanto simétrica como de clave pública.



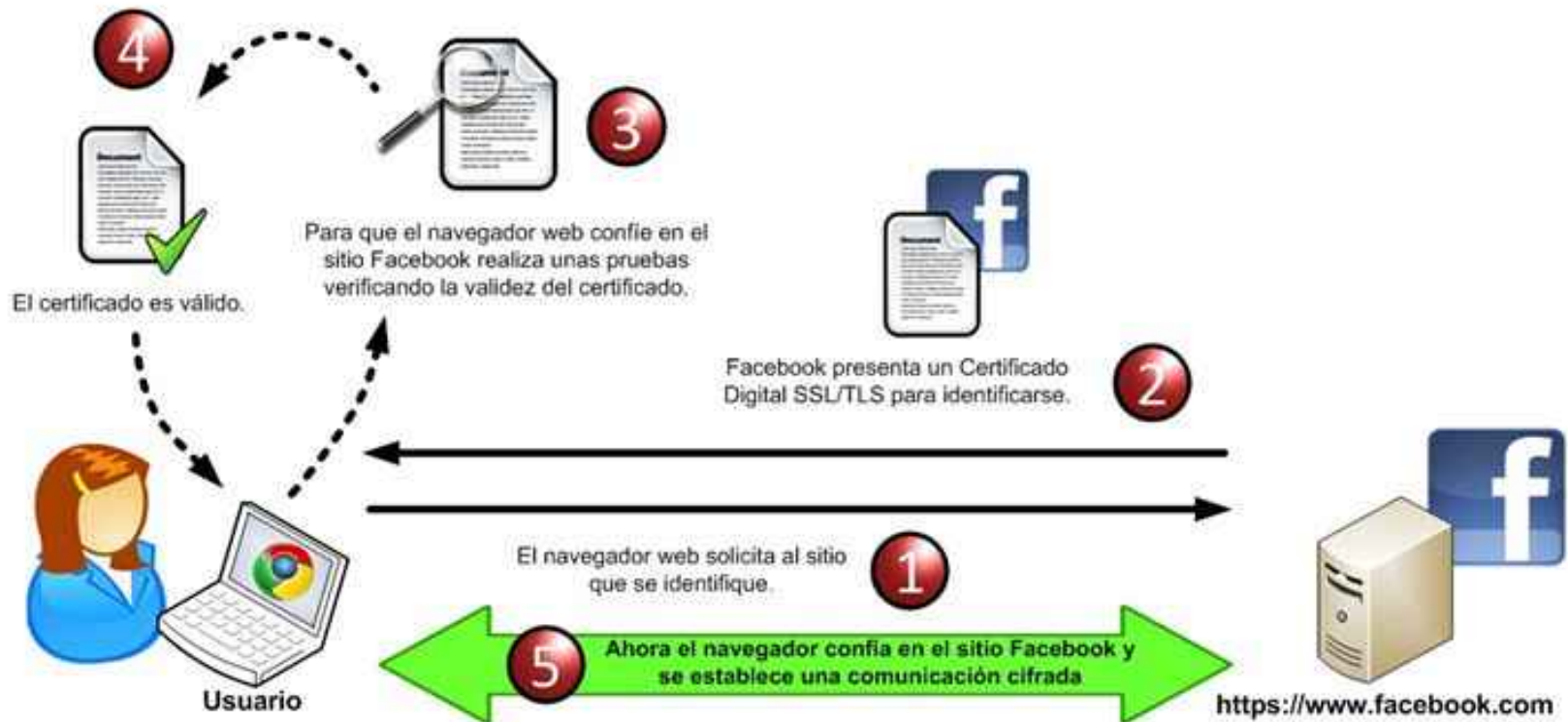
3.3 CARACTERÍSTICAS

- **Confidencialidad:** capacidad de garantizar que la información solo podrá ser accesible por aquellos a quien va dirigida.
- **Integridad:** capacidad de asegurar que los datos no serán modificados durante la transmisión.
- **Autenticación:** Garantiza que el interlocutor es quien dice ser.



3.4 FUNCIONAMIENTO

- Ejemplo petición



3.4 FUNCIONAMIENTO

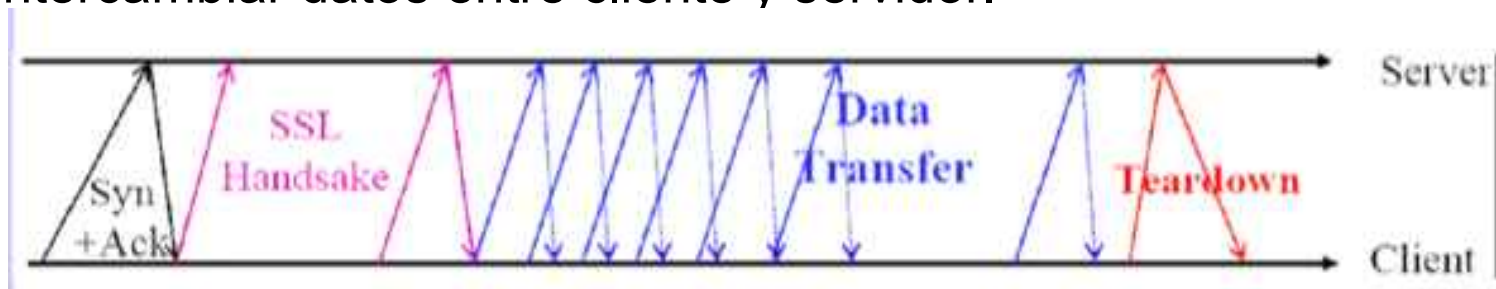
- **Confidencialidad**

Protocolo de Handshake

- Utiliza **criptografía de clave pública** para establecer una clave compartida entre cliente y servidor, y se negocian los algoritmos de cifrado y mantenimiento de la integridad que controlarán la conexión.
 - El cifrado asimétrico afecta al rendimiento (**overhead**).

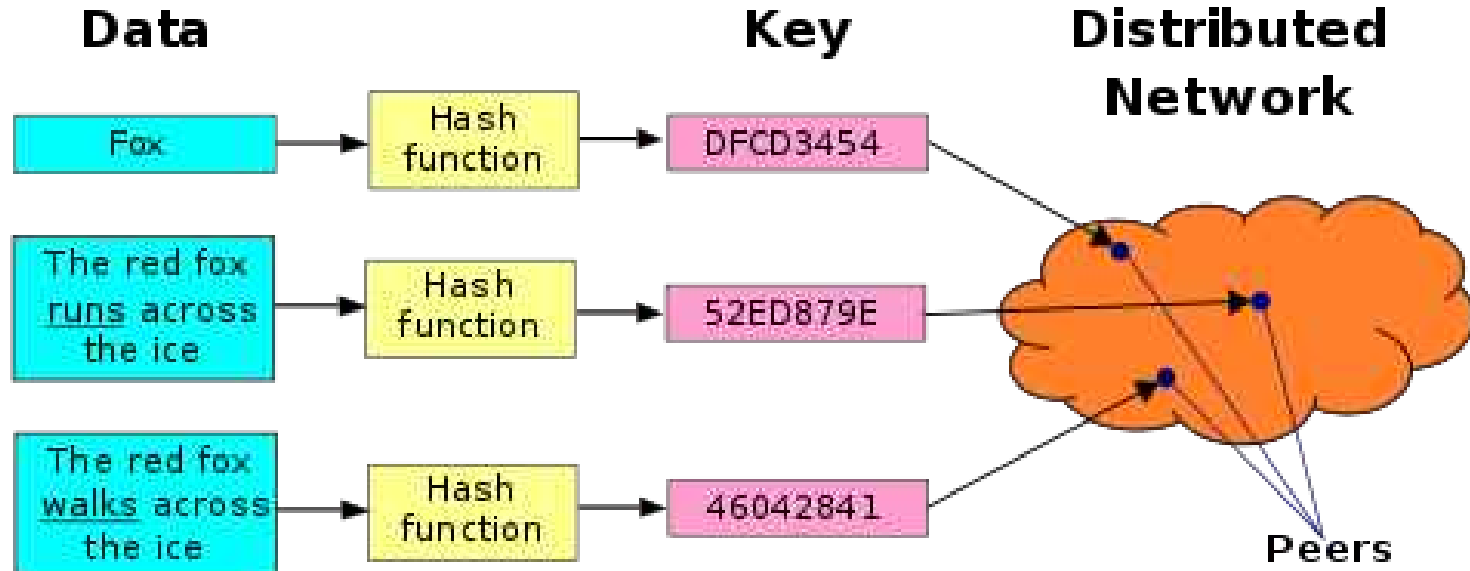
Protocolo de transferencia

- Utiliza la clave compartida establecida en el punto anterior para intercambiar datos entre cliente y servidor.



3.4 FUNCIONAMIENTO

- **Integridad**
 - TLS proporciona integridad de los mensajes enviados mediante el cálculo de un resumen o hash de mensaje. El algoritmo es consensuado durante la fase de handshake.



3.4 FUNCIONAMIENTO

- **Autenticación**

Hasta ahora hemos cifrado las conexiones però... ¿qué pasaría si un tercero intercepta la primera comunicación y se hace pasar por nosotros y por el banco?



3.4 FUNCIONAMIENTO

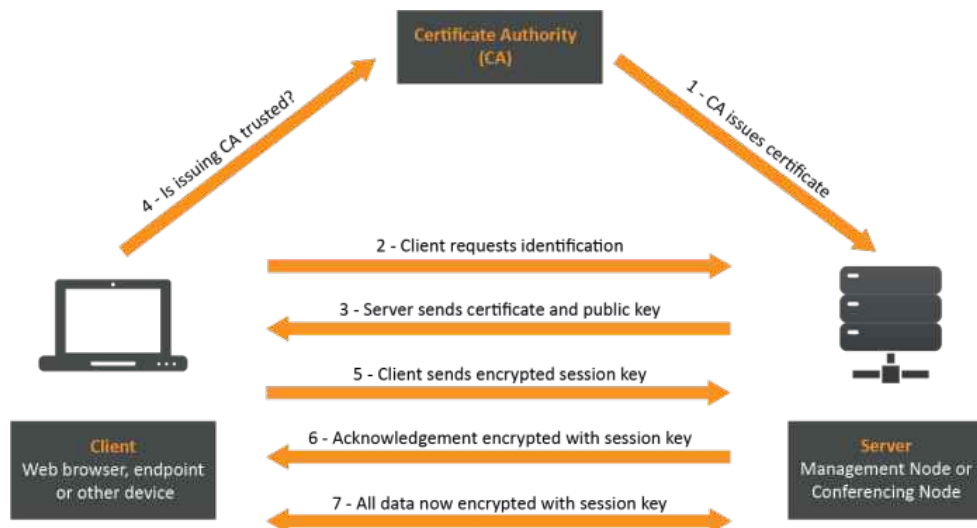
- **Autenticación**

- **Ana** utiliza la clave pública que piensa que es de **Pepe** para cifrar el mensaje.
- **Pepe** utiliza la clave pública que piensa que es de **Ana**.
- **Man** no solo accede a la información sino que puede modificarla (Ex. transferir dinero a otra cuenta)



3.4 FUNCIONAMIENTO

- **Autenticación**
 - **Solución: Certificados y autoridades de confianza**
 - **Un tercero verifica la autenticidad e identidad de los certificados** y de la información que contienen mediante su firma.
 - **Chain of Trust** → cadena de confianza

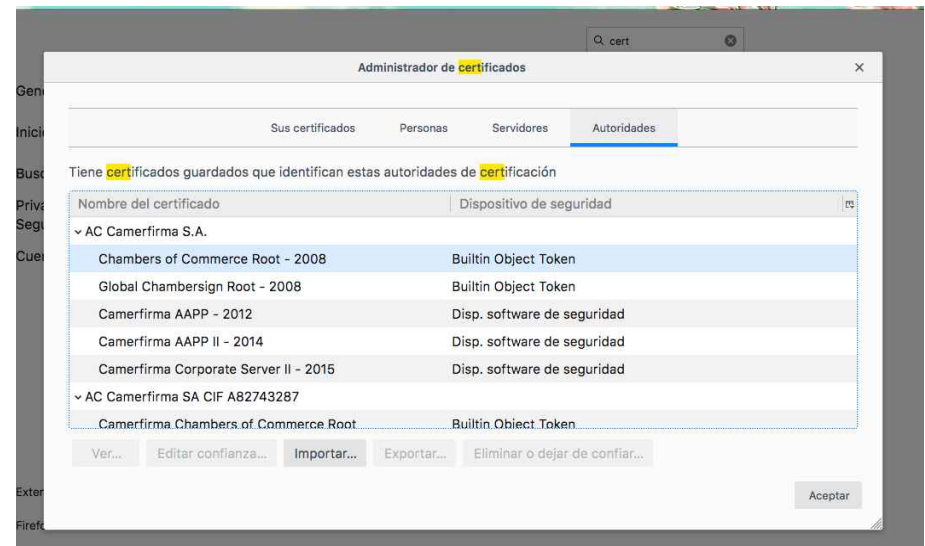
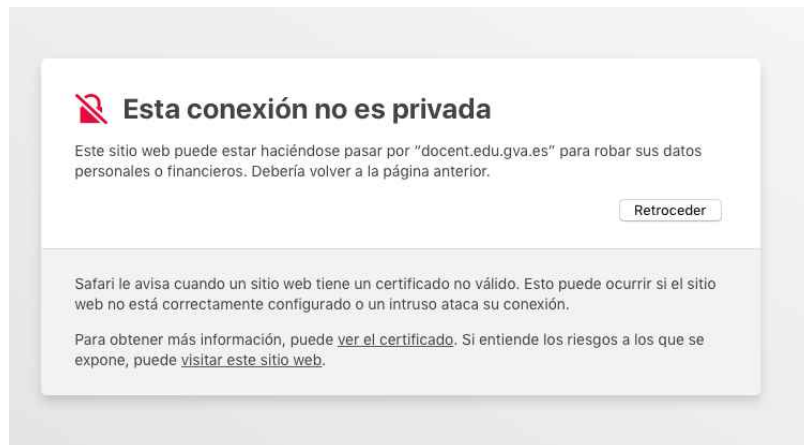


4. CERTIFICADOS Y AUTORIDADES DE CERTIFICACIÓN

- **Autoridad certificadora (CA).** Se trata de una tercera entidad de confianza, responsable de emitir y revocar certificados digitales.
- **Certificados:** recogen ciertos datos de su titular y su clave pública, y están firmados electrónicamente por la Autoridad certificadora mediante su clave privada.
 - Certifica que una clave pública pertenece a su propietario

4. CERTIFICADOS Y AUTORIDADES DE CERTIFICACIÓN

- Los clientes disponen de las claves públicas de aquellas autoridades de certificación en las que confían.
 - Cualquier certificado que no haya sido firmado por una CA de confianza por el navegador, emitirá un mensaje de error.



4. CERTIFICADOS Y AUTORIDADES DE CERTIFICACIÓN

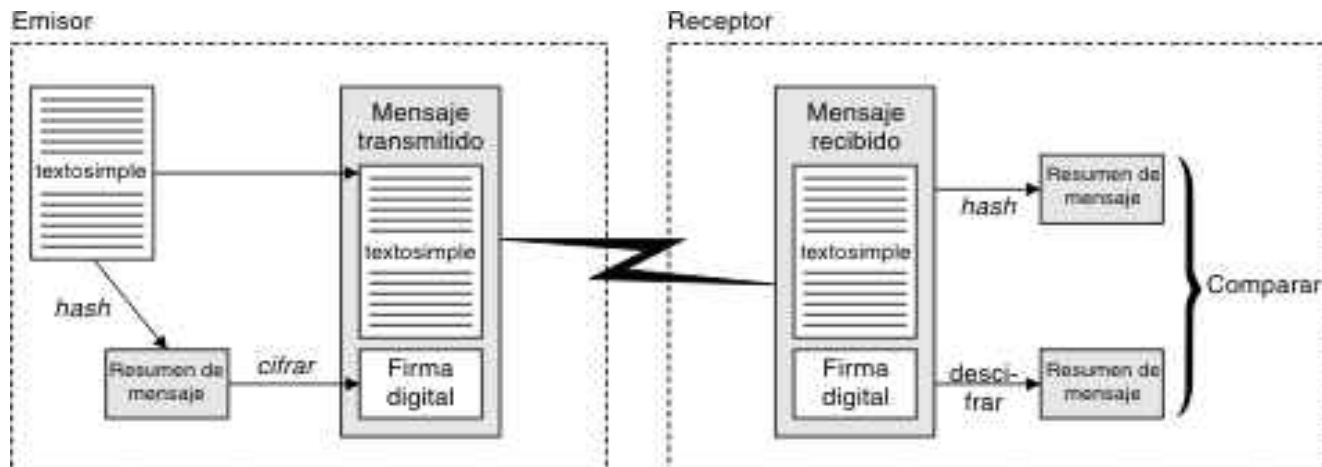
"The root certificates distributed with common browser software are added according to criteria defined by the browser supplier and vary from "pay us lots of cash"

"Los certificados raíz distribuidos en los navegadores más comunes son añadidos de acuerdo a criterios propios del creador y pueden variar de acuerdo a la cantidad que pagan"

4. CERTIFICADOS Y AUTORIDADES DE CERTIFICACIÓN

- **Funcionamiento**

- Si el navegador es capaz de "descifrar" la firma del mensaje mediante la clave pública de la CA en la que confía y, además ésta coincide con el hash calculado a partir de la información presente en el certificado, podrá validar la autoría del servidor.



4. CERTIFICADOS Y AUTORIDADES DE CERTIFICACIÓN

- Tipos de certificados
 - **Validación del dominio (DV):** Verifica solo si el dominio está registrado a nombre de quien pidió el certificado (nivel de seguridad bajo).
 - **Validación de la organización (VO):** Se investiga si la organización es propietaria del dominio. (nivel medio).
 - **Validación extendida (VA):** Validación oficial de la entidad; registros oficiales, uso que se hará del dominio,... (nivel alto)

4. CERTIFICADOS Y AUTORIDADES DE CERTIFICACIÓN

- Tipos de certificados



Validación del Dominio

Ventajas

- Se emite instantáneamente (menos de 10 minutos)
- Bajo costo, puesto que la validación es automática
- Cifrado básico
- Seguridad rápida y simple
- Garantía incluida

Desventajas

- Prueba sólo que su sitio es seguro (no su empresa)
- No otorga confianza a su negocio (ya que su negocio no está controlado)

Uso sugerido

- Solo para pruebas y uso interno
- Todas aquellas personas que necesitan un cifrado básico



Validación de la Organización

Ventajas

- Validación de sitio web y de su empresa
- Prueba que su negocio es legítimo y que usted es el propietario o está autorizado a ejecutarlo
- Verificación humana
- Licencia de servidor ilimitado
- Incluye sellos de sitio seguro

Desventajas

- La emisión del certificado puede requerir hasta dos días, si bien hacemos todo lo posible para emitirlo en el día
- Levemente más costoso del DV a causa de la investigación humana

Uso sugerido

- Sitios de comercio online
- Todas las personas que deseen demostrar que sus sitios y sus negocios son confiables



Validación Ampliada

Ventajas

- Activa la barra de direcciones verde
- Inspira los más altos niveles de confianza en sus clientes
- Protege su sitio contra el phishing
- Asegura los directores y demás personas interesadas de su empresa
- Procedimientos de investigación rigurosos

Desventajas

- Más caro
- Tómese hasta 5-10 días para publicar

Uso sugerido

- Sitios de comercio online
- Marcas nacionales y globales
- Todo negocio que desee impulsar sus ventas
- Toda persona que desee infundir más confianza a sus visitantes online
- Para una máxima protección contra el phishing

4. CERTIFICADOS Y AUTORIDADES DE CERTIFICACIÓN

- **Autoridades de certificación de pago**
 - Un criterio de selección de la autoridad es el nivel de aceptación que tienen los navegadores de ella.
 - Todos los navegadores disponen de una lista de certificados aceptados.
 - Son caros, y renovables anualmente.



4. CERTIFICADOS Y AUTORIDADES DE CERTIFICACIÓN

- **Certificados auto-firmados**

- Si nuestros clientes son internos, podemos ser nosotros nuestra propia autoridad certificadora
 - Certificados auto-firmados
- Tendremos que instalar la clave pública en los navegadores de los clientes para no recibir el mensaje de error correspondiente



4. CERTIFICADOS Y AUTORIDADES DE CERTIFICACIÓN

- **Let's encrypt**
 - **Autoridad de certificación gratuita** promovida por las principales compañías que ofrecen servicios en Internet.
 - **Aceptada** por la mayor parte de los navegadores web como "*autoridad de confianza*"
 - Proporciona 2 tipos de certificados
 - **SSL Individual**: Cubre un dominio.
 - **SSL Wildcard**: Cubre todos los subdominios.

4. CERTIFICADOS Y AUTORIDADES DE CERTIFICACIÓN

- **Let's encrypt**
 - Renovable cada **90 días**.
 - El proceso de renovación es automático mediante un script que se ejecuta en el servidor.

