# Ciberseguridad

## Grado en Ingeniería Informática

## M2 - Criptografía, Identificación y Control de Accesos

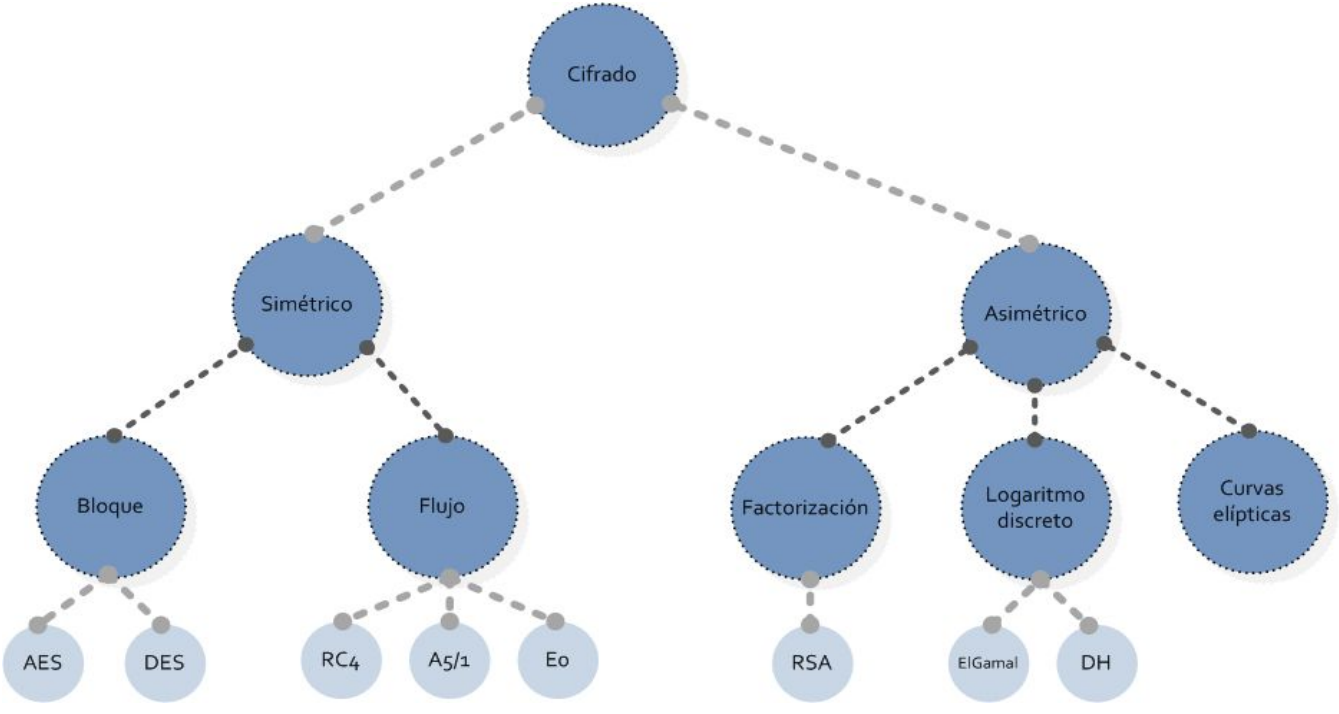
Oscar Delgado
oscar.delgado@uam.es

Álvaro Ortigosa (Coord.)
alvaro.ortigosa@uam.es

UAM Universidad Autónoma de Madrid

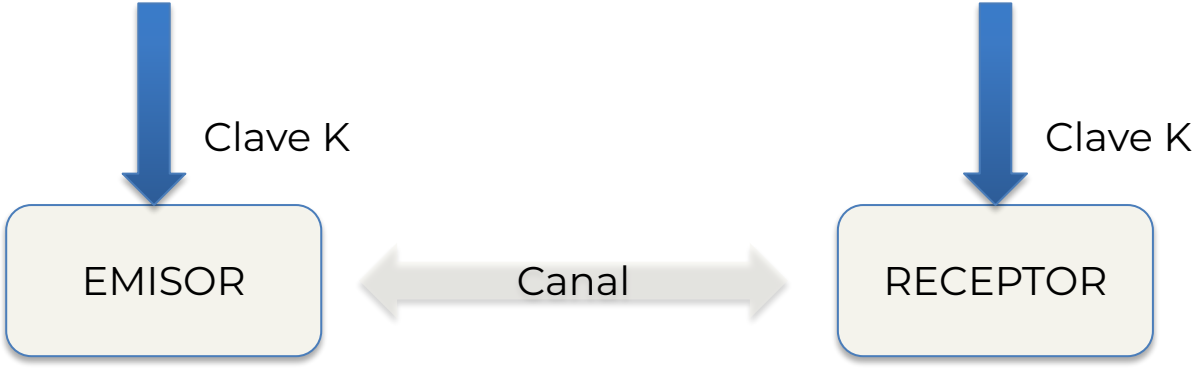
# CIFRADO ASIMÉTRICO
## Repaso

# Taxonomía


Cifrado
Simétrico
Asimétrico
Bloque
Flujo
Factorización
Logaritmo discreto
Curvas elípticas
AES
DES
RC4
A5/1
E0
RSA
ElGamal
DH

# Cripto simétrica

## Clave simétrica



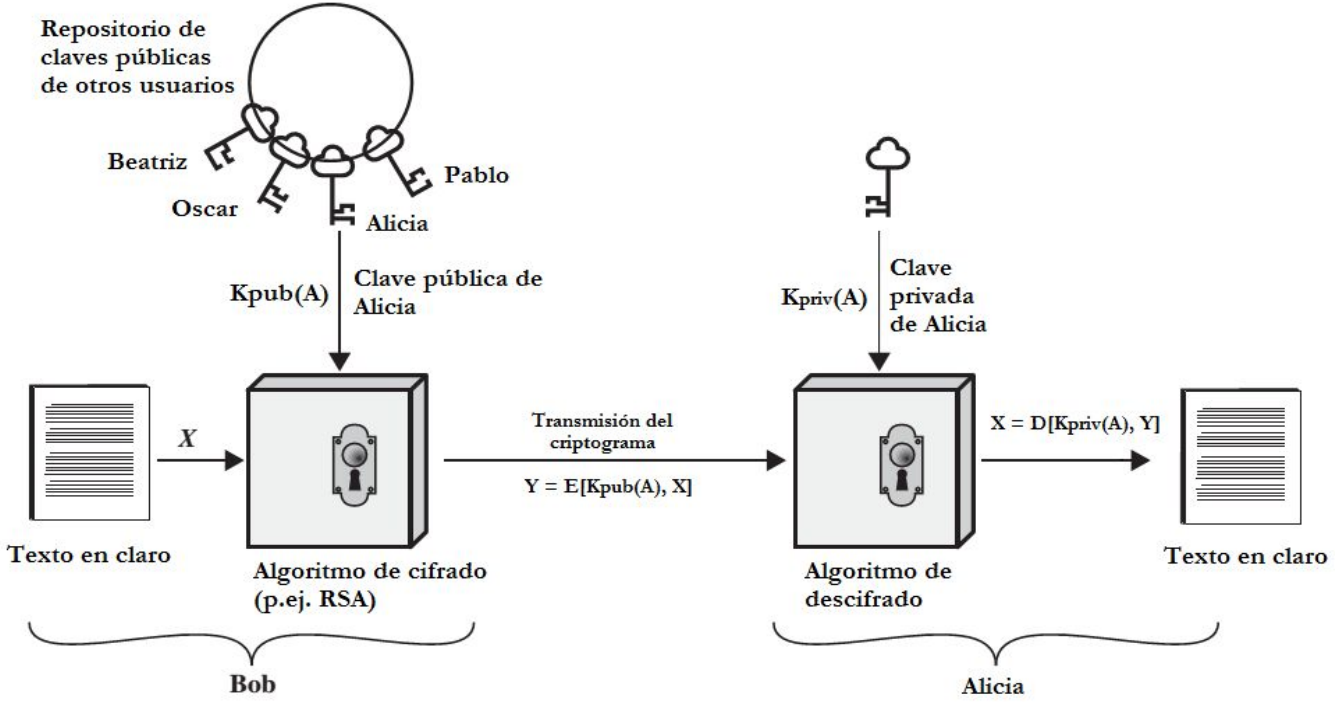

**Inconveniente:** distribución de claves

# Elementos básicos

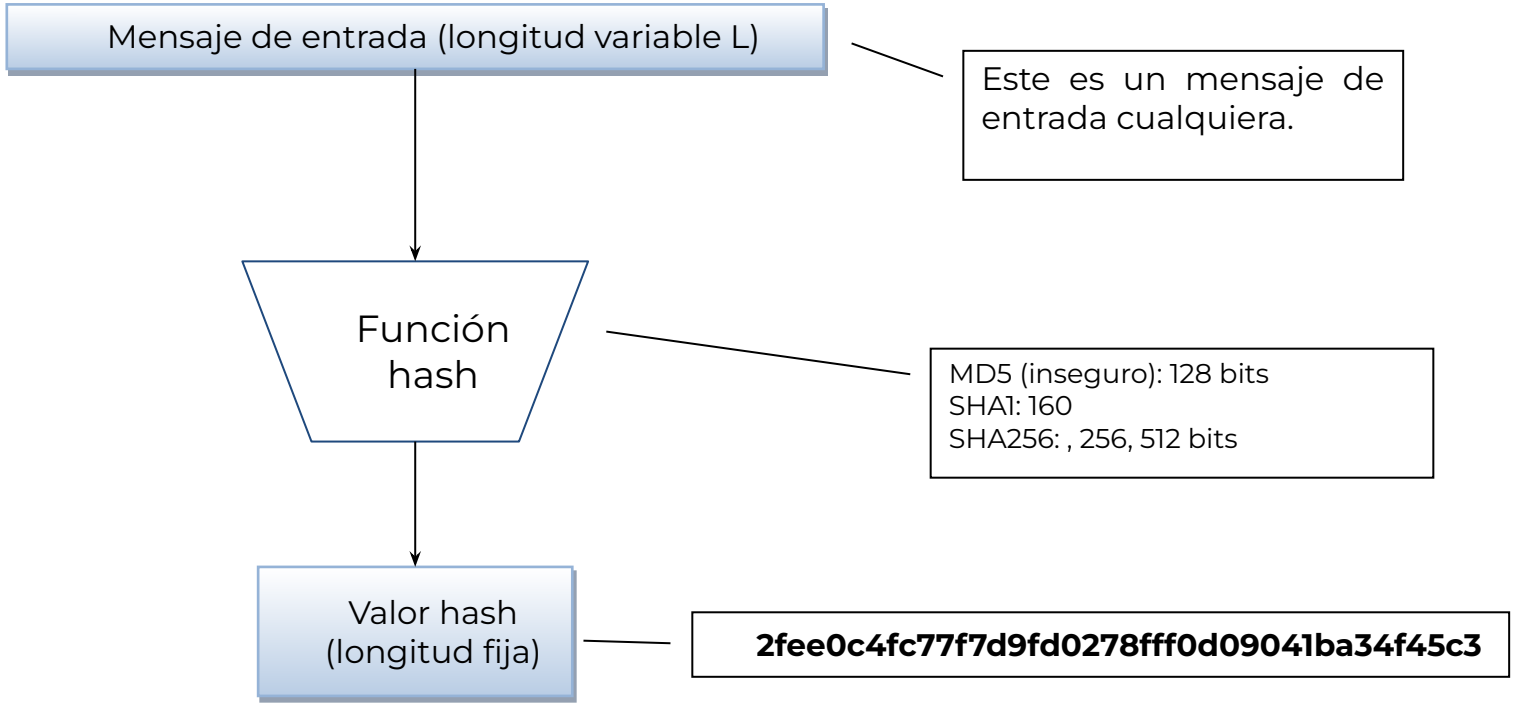
- Dos claves, denominadas clave pública y privada, vinculadas matemáticamente.

Principio básico:
"Lo que cifra con una de las claves, sólo se puede descifrar con la otra"

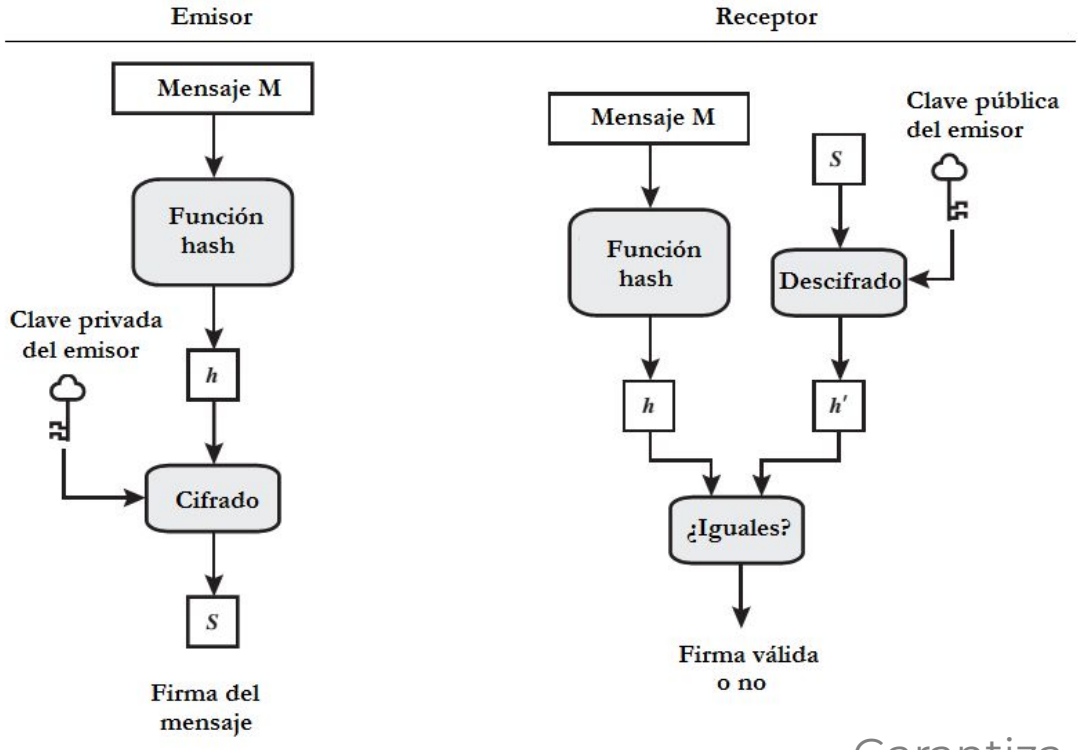
# Elementos básicos



# Funciones hash

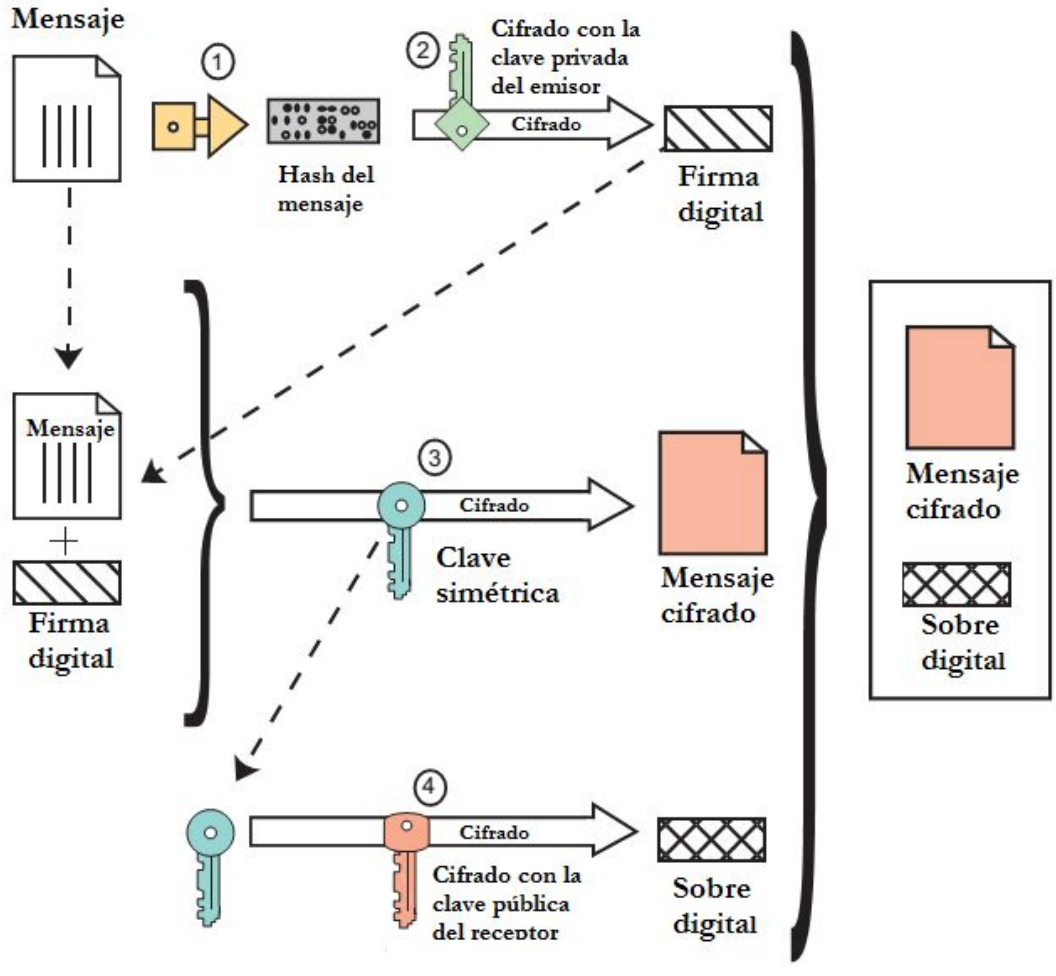


# Firma digital

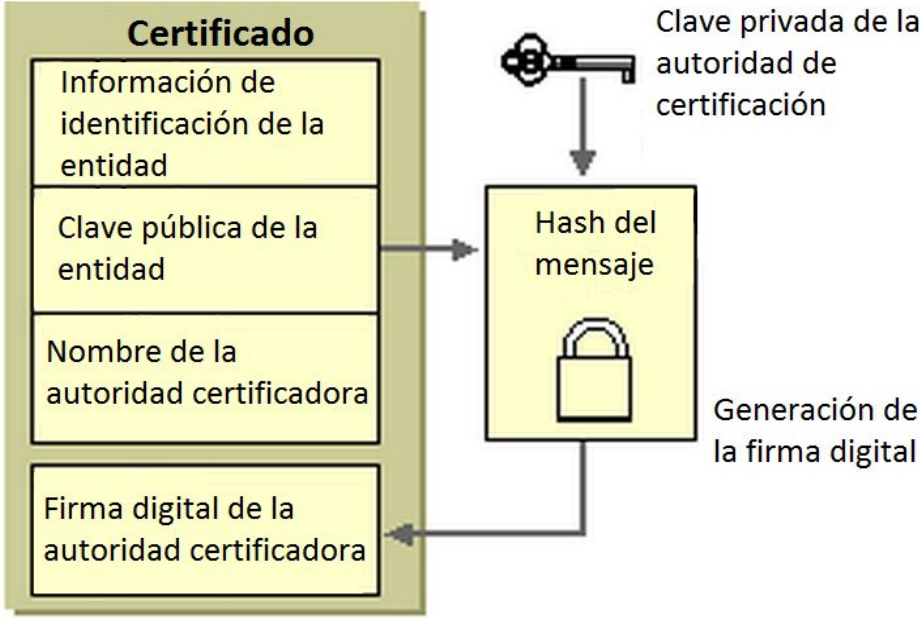



Garantiza el no repudio

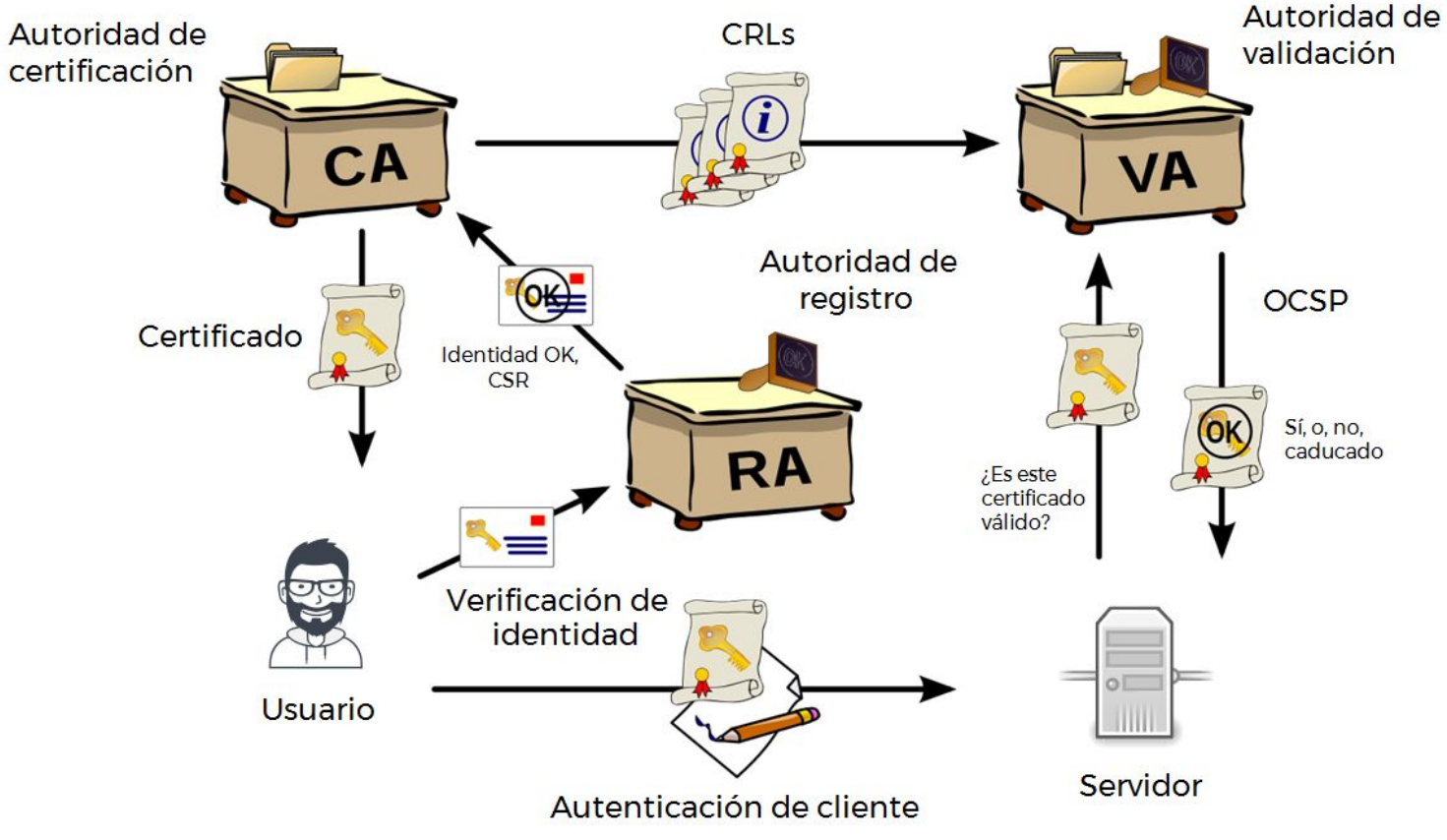
# Esquemas híbridos



# Certificados digitales



# PKI – Public Key Infraestructure

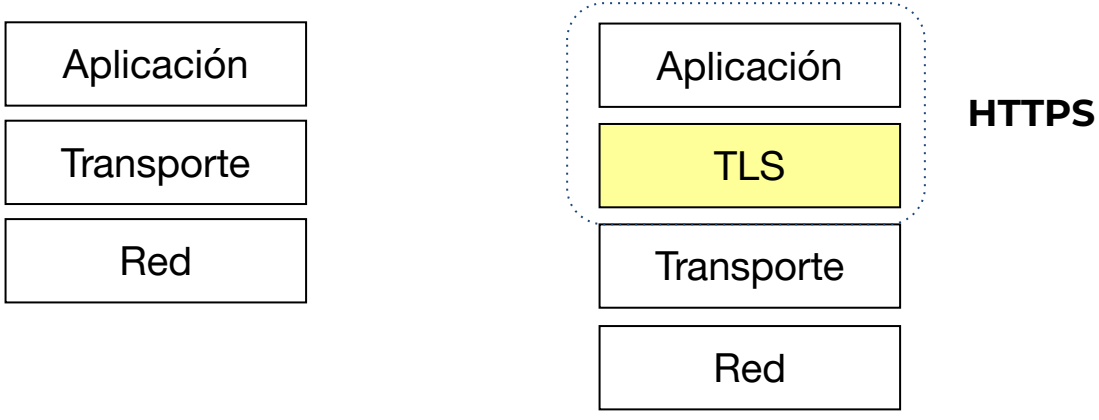


# Seguridad Web // TLS

# Transport Layer Security

- Protocolo que aporta **secreto**, **autenticidad** e **integridad** a las comunicaciones, típicamente utilizado sobre HTTP (HTTPS)
- TLS evolucionó sobre *Secure Sockets Layer* (SSL), desarrollado inicialmente por Netscape.
- Son protocolos diferentes, pero aún se suelen utilizar SSL como referencia genérica.

# TLS // Protocolo de transporte




- TLS es un protocolo de transporte que puede proteger (casi) cualquier aplicación
- Es necesario que la app se adapte a TLS (no son *sockets* normales)

# Negociación TLS

El proceso comienza con una negociación (*handshake*) C-S en la que:

1. Se especifica qué versión TLS usarán (TLS 1.0, 1.2, 1.3, etc.)
2. Deciden qué suite criptográfica soportan
3. El cliente autentica al servidor utilizando su certificado
4. Generan **claves de sesión** para cifrar las comunicación cuando acabe el *handshake*

# Suite criptográfica

- Algoritmo de clave pública (RSA, EC)
- Algoritmo de clave secreta (AES)
- Algoritmo de hashing

- Es importante tener la oportunidad de "negociar", por si se descubre alguna vulnerabilidad en algún protocolo

- Negociación: el cliente ofrece opciones y servidor elige una (puede rechazar la conexión si todas son inseguras)

**Algoritmos simétricos habituales**

- AES – Advanced Encryption Standard
- ChaCha - cifrado en flujo

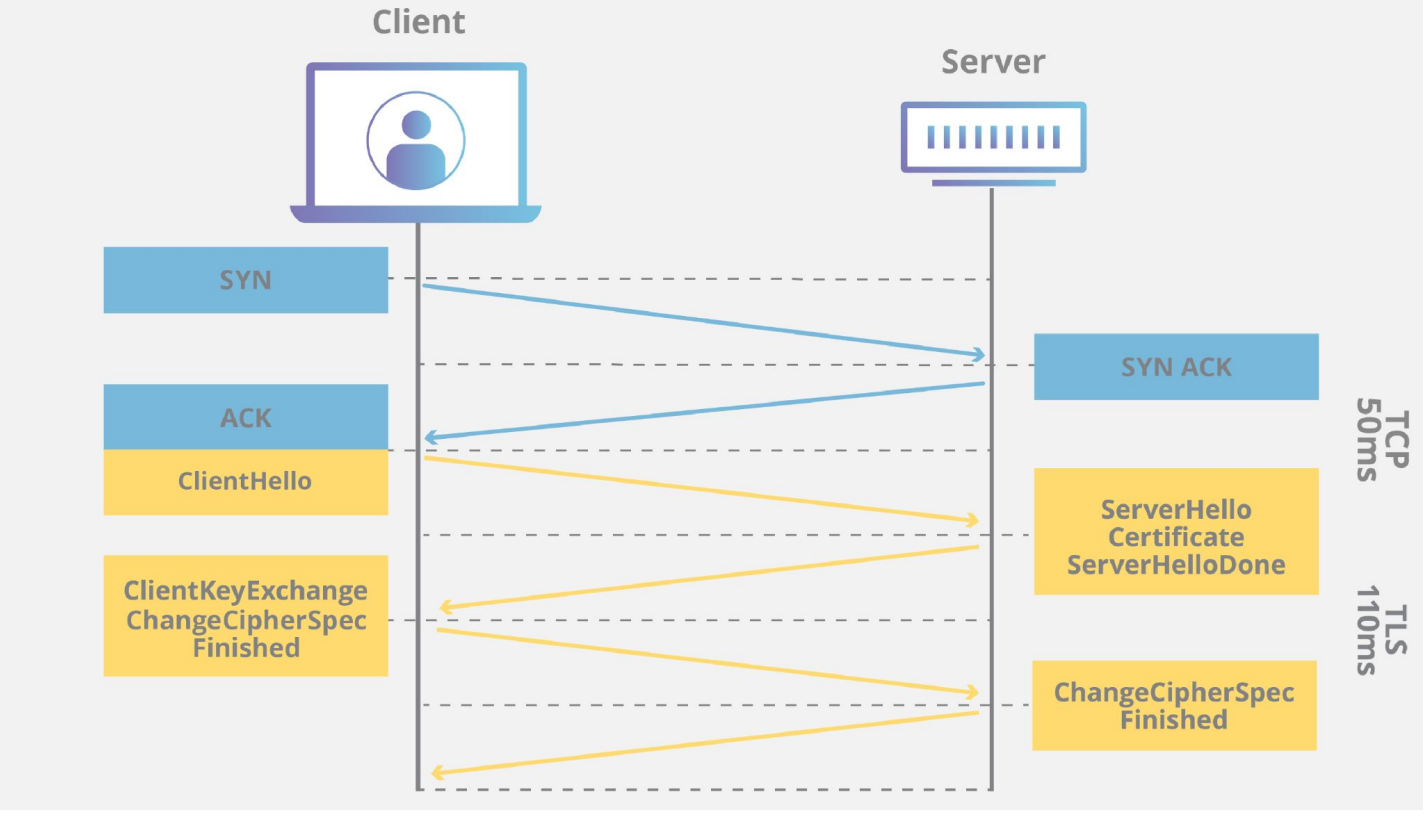
**Clave pública**

- Curvas elípticas (RSA se va abandonando porque son más lentas y posible riesgo del computador cuántico)

**Hash**

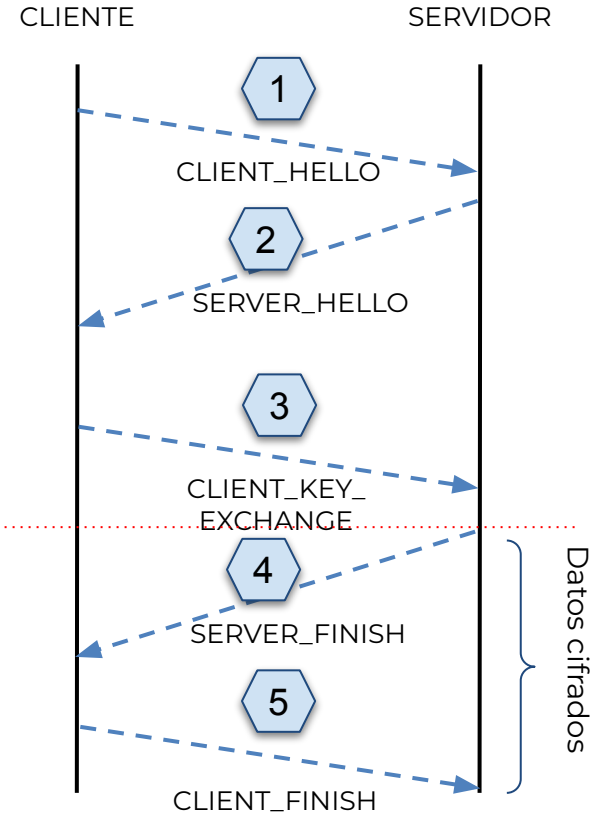
- Solo familia SHA2 y SHA3

# Handshake // Alto nivel




**Fuente**: Cloudfare

# Handshake // Detalle



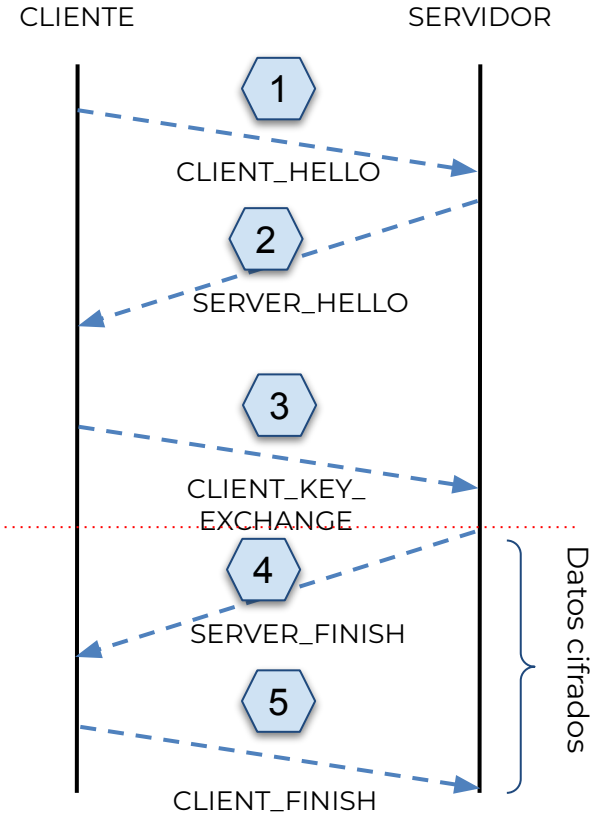

1. **CLIENT_HELLO**: El cliente envía la versión de TLS que quiere utilizar, su suite criptográfica y un número aleatorio (nonce), $n_c$
2. **SERVER_HELLO**: El servidor envía su certificado, los algoritmos elegidos, su propio nonce ($n_s$) y el nonce del cliente cifrado con su clave privada
3. **CLIENT_KEY_EXCHANGE**: el cliente:
   a. Comprueba el certificado del servidor
   b. Genera el *pre_master_secret* (un número aleatorio de 48 bytes) y lo cifra con la clave pública del servidor
4. **SERVER_FINISH**: el servidor deriva la **clave de sesión** (*master_secret*):

$$master\_secret = hash(pre\_master\_secret \,||\, \text{"master secret"} \,||\, n_c + n_s)\ [0..47]$$

y utiliza solo los primeros 48 bytes para cifrar la comunicación a partir de este punto con el algoritmo simétrico elegido. Añade un MAC de todos los mensajes anteriores

5. **CLIENT_FINISH**: deriva la misma **clave de sesión** y añade su MAC de los mensajes anteriores

# Handshake // Detalle




- Los **nonces** son números aleatorios, que sirven para evitar el **ataque de repetición** (*replay attack*): un atacante captura los mensajes intercambiados y simplemente los repite en un momento posterior

- Los mensajes 4 y 5 “firman” todo el proceso y los vinculan con los nonces utilizados

- El mensaje CLIENT_KEY_EXCHANGE es un tipo sencillo de primitiva llamada **Transporte de claves**

ACTIVIDAD

# TLS handshake

Vamos a ver la negociación TLS en acción. Para ello, podemos utilizar el comando ‘openssl’

```
# openssl s_client -connect amazon.es:443
```

Podemos ver más detalle, cómo evoluciona el estado de cliente y servidor con el flag -state:

```
# openssl s_client -state -connect amazon.es:443
```

**Preguntas**

1. ¿Pues la versión TLS finalmente negociada? ¿Quién la ha elegido?
2. Observa los campos Cipher y Master-key

# Sellado de tiempo

# Sellado de tiempo

- Permite probar que un dato *existía en un momento dado del tiempo*, y que no ha sido modificado desde entonces

- Esencial para la **firma digital**, pues ésta, *per se*, no contiene información sobre el momento de su creación

# Sellado de tiempo

- Idealmente, la marca de tiempo debe ser generada por una parte diferente a quien realiza la firma: *Time Stamp Authority* (TSA)

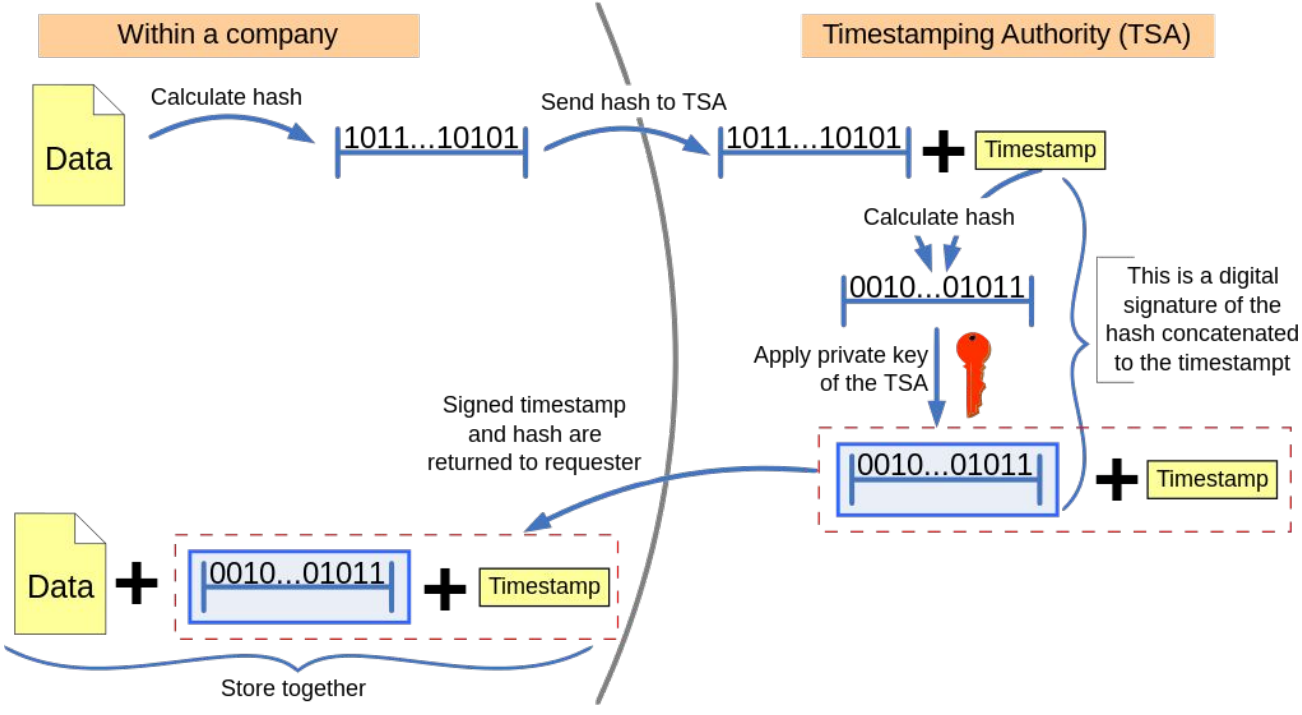
- La FNMT es también TSA, utilizando una fuente de tiempo segura: la del Real Observatorio de la Armada, fuente oficial de la hora legal en España (protocolo NTP)

# Sellado de tiempo // Aplicaciones

| Firmas digitales | Notaría digital | Análisis forense |
| --- | --- | --- |
| • Aumenta su seguridad, al incluir referencia temporal | • Prueba de existencia de unos datos en un momento del tiempo<br>• Protección de copyright | • Cadena de custodia de logs, ficheros, etc.. |

# Sellado de tiempo

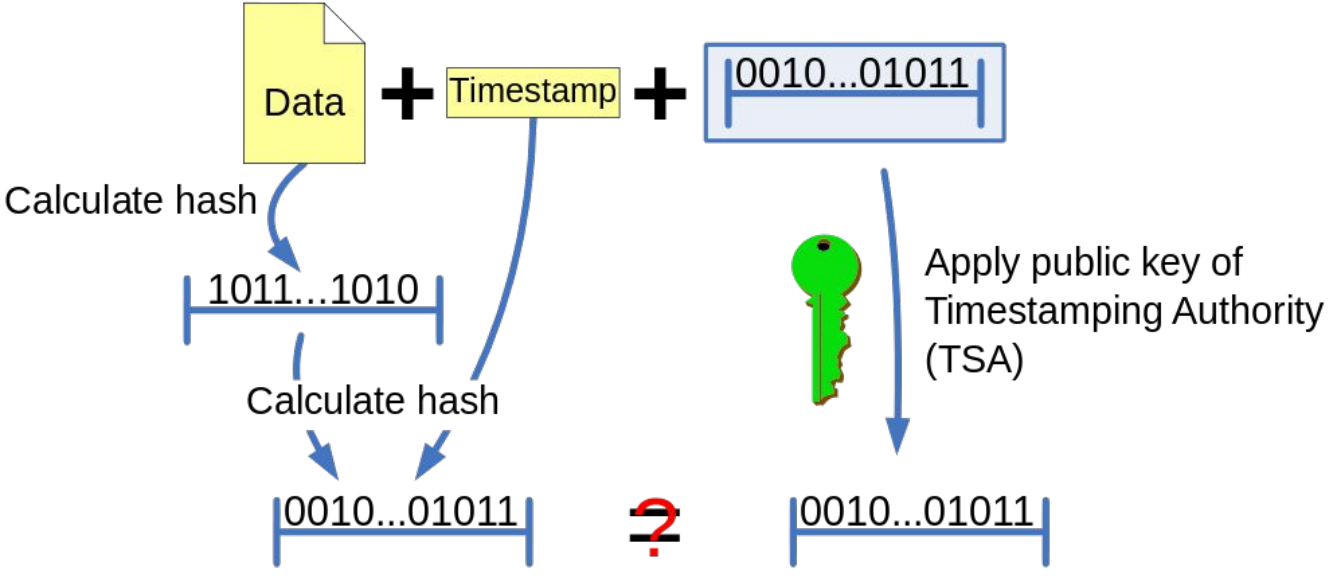
## Generación




**Fuente**: Cryptography and Network Security, William Stallings

# Sellado de tiempo

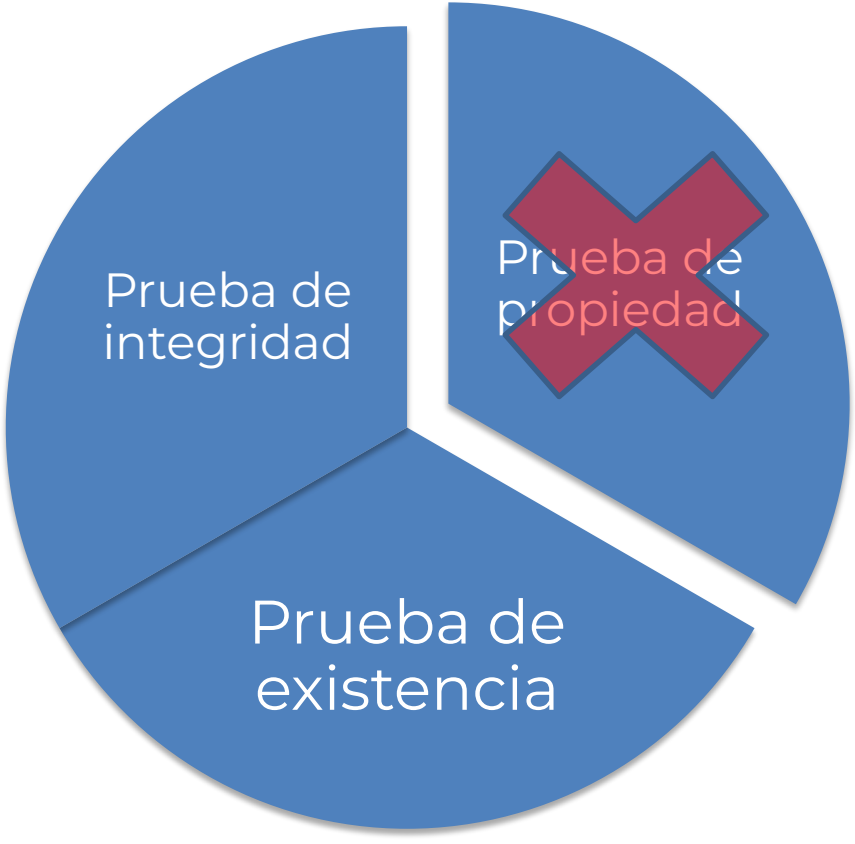
## Verificación




**Fuente**: Cryptography and Network Security, William Stallings

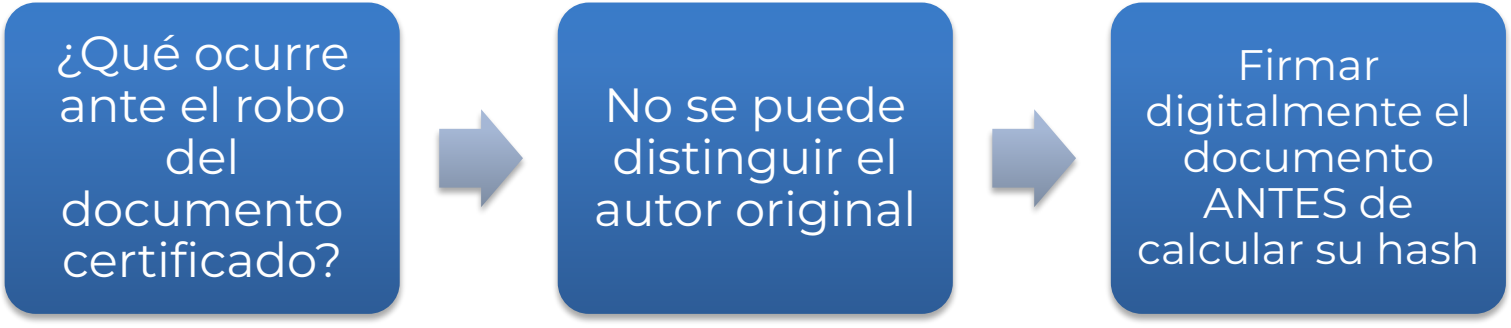
# Sellado de tiempo

Limitaciones



# Sellado de tiempo

## Prueba de propiedad



ACTIVIDAD

# Sellado de tiempo

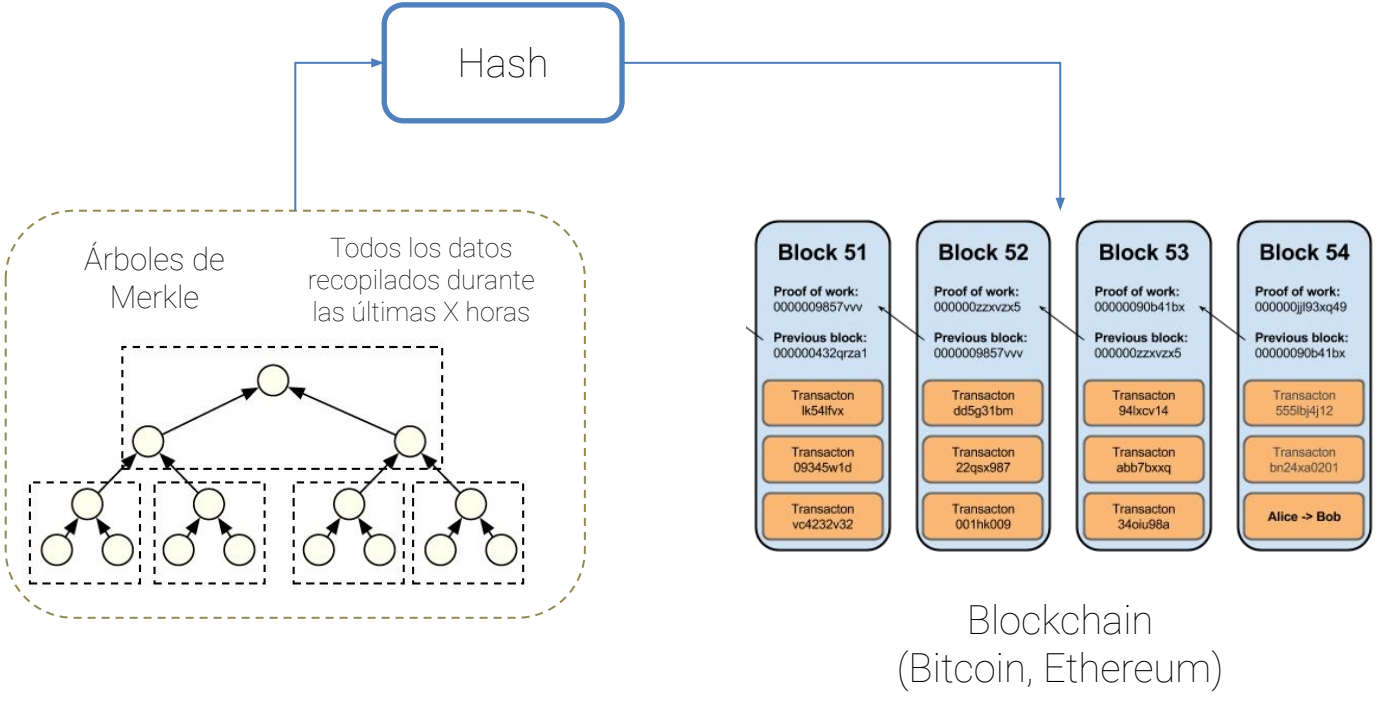
Comando de Linux para conocer el estado de los sockets y conexiones del SO:

```
# netstat -a
```

**Preguntas**

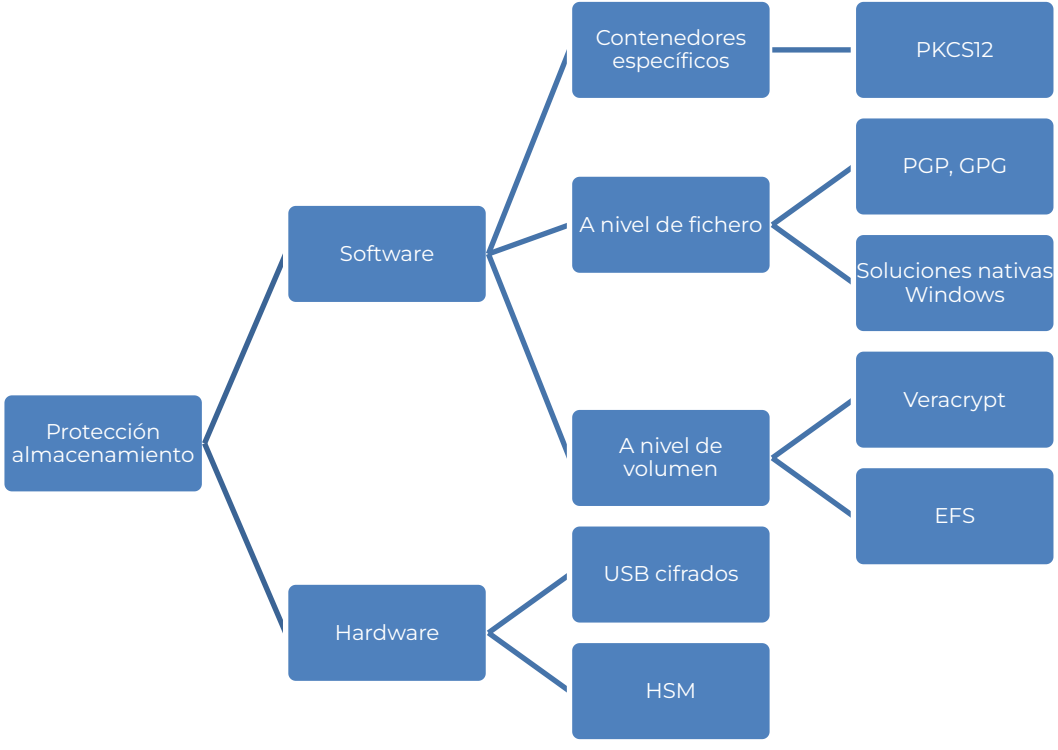
1. ¿Cuántos están en estado LISTEN y ESTABLISHED?
2. ¿Qué diferencia hay entre un socket escuchando en localhost y 0.0.0.0?

# Certificación basada en *blockchain*



# Almacenamiento seguro de certificados y claves

# Protección de claves y certificados



# Protección por software

## Ventajas

Utilizar contenedores criptográficos específicos, como PKCS#12

Se pueden utilizar tanto para unidades de almacenamiento interno como externo, como USBs o discos duros

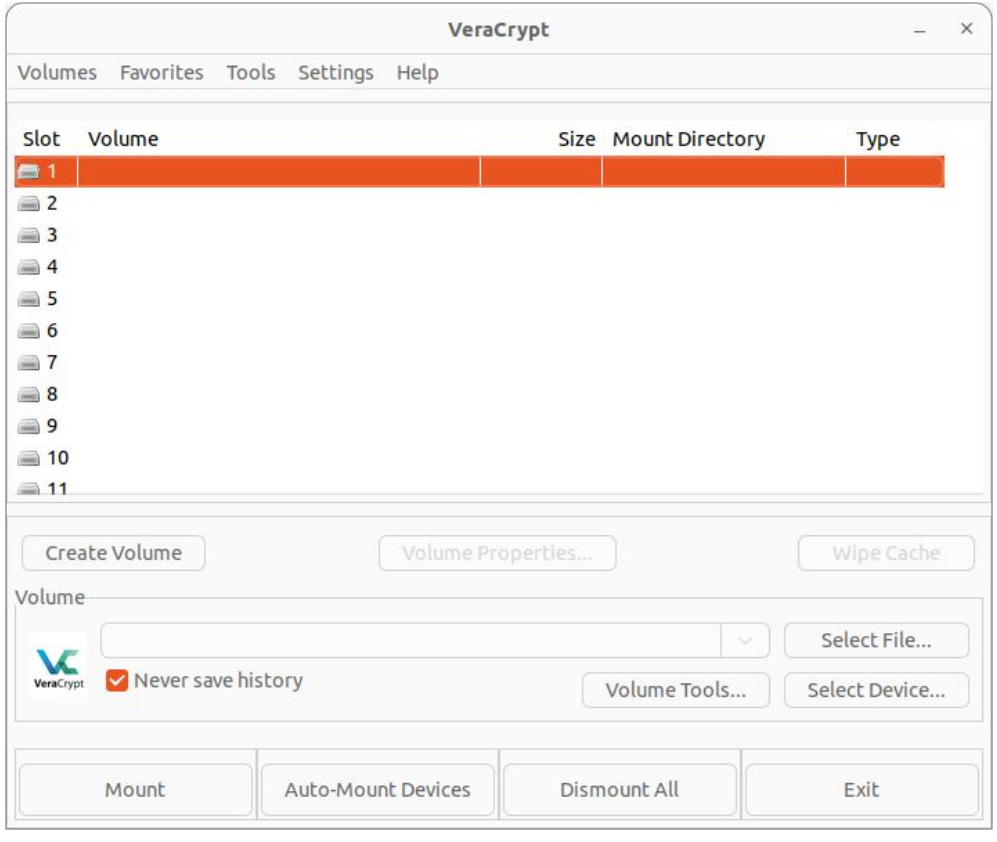
# Protección por software

## Inconvenientes

Necesario instalar software en los ordenadores desde los que se desee almacenar información “segura” y recuperarla

Dependen de un único secreto (contraseña) para su protección

# Veracrypt


VeraCrypt
Volumes
Favorites
Tools
Settings
Help
Slot
Volume
Size
Mount Directory
Type
1
2
3
4
5
6
7
8
9
10
11
Create Volume
Volume Properties...
Wipe Cache
Volume
Select File...
Never save history
Volume Tools...
Select Device...
Mount
Auto-Mount Devices
Dismount All
Exit

# VeraCrypt

- **Unidades virtuales:**
  - Generadas por el programa bajo demanda del usuario
  - Se comportan exactamente igual que los discos duros
  - Pueden ser montadas y desmontadas por el usuario

- **Contenedor:**
  - Tipo de fichero creado por el programa que pueden ser accedidos a través de una unidad virtual
  - Contienen información cifrada

# VeraCrypt

- El *driver* de TrueCrypt procesa todas las peticiones que se realizan sobre una unidad virtual
- Todas aquellas peticiones que se realizan sobre información “no sensible” son realizadas directamente por el sistema operativo

# VeraCrypt // Pre-boot Authentication

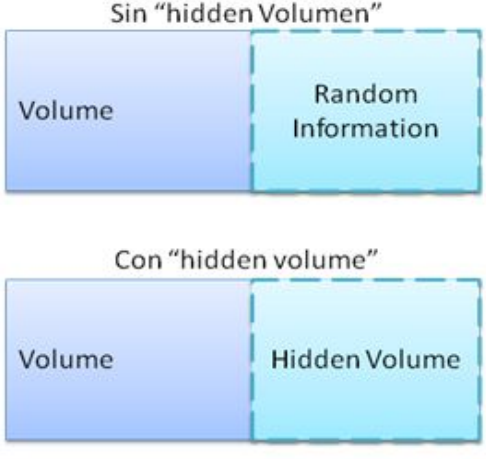
- Cifrado de todo el volumen en el que se encuentra un sistema instalado

- Alto nivel de seguridad ya que todos los archivos del sistema, como temporales, ficheros de hibernación, información sobre las aplicaciones ejecutadas, nombres y localizaciones de ficheros, etc., se mantienen cifrados

- VeraCrypt utiliza su propio Boot Loader para autenticar a los usuarios antes de la carga del sistema.

# VeraCrypt // Volumen oculto

- Residen dentro de volúmenes convencionales, de forma que a simple vista no pueden ser detectados.
- Cuando se crea un volumen, TrueCrypt rellena el espacio escogido para ese volumen, con información aleatoria.
- La clave utilizada para este segundo volumen escondido debe ser completamente diferente a la utilizada para el volumen convencional.



ACTIVIDAD

# Veracrypt

Commando de Linux para conocer el estado de los sockets y conexiones del SO:

```
# netstat -a
```

**Preguntas**

1. ¿Cuántos están en estado LISTEN y ESTABLISHED?
2. ¿Qué diferencia hay entre un socket escuchando en localhost y 0.0.0.0?

# VeraCrypt

## Ejercicio MC_VERACRYPT

- Tiempo de realización: 15 minutos
- Individual

# Protecciones hardware



Wallets específicas

# Soluciones hardware

✔ Ventajas

- Generalmente no necesitan de ninguna instalación
- Se conectan como unidades de almacenamiento convencionales

X Desventajas

- Menos flexible, ya que el software puede ser instalado en distintos tipos de dispositivos

# Hardware Security Modules (HSM)

- HSM gama de dispositivos que ofrecen gran seguridad a la información
- Seguridad a nivel físico y lógico

# HSM – Medidas antimanipulación

- La seguridad de estos dispositivos se basa en sus extremas medidas físicas antimanipulación.
- Pueden clasificarse, básicamente, en medidas:
  - **Pasivas**: tratan de dificultar la inspección y manipulación de los componentes activos y las claves almacenadas.
  - **Activas**: detectan los intentos de intrusión e, inmediatamente, destruyen todas las claves criptográficamente sensibles.

# HSM – Medidas antimanipulación pasivas

- La más sencilla es una simple carcasa de acero.
- Útil sólo si se combina con seguridad física de acceso (a un CPD, por ejemplo).
- El peso puede suponer también una medida disuasoria y algunos HSM incluyen peso extra a propósito.

# HSM – Medidas antimanipulación pasivas

- Encapsulado de todo el dispositivo en material antimanipulación (resina epoxy):

# HSM – Medidas antimanipulación activas

- Algunas de las medidas activas incluyen:
  - **Sensores térmicos**: protegen contra ataques que impliquen enfriamiento. El enfriamiento extremo provoca efectos de remanencia de datos en memorias RAM.
  - **Sensores de rayos X** : detectan este tipo de radiación y borran las claves antes de que éstas puedan ser “quemadas” en la RAM por la radiación.

# HSM – Medidas antimanipulación activas

- **Sensores de luz**: borran las claves cuando detectan luz visible. Pueden ser engañados trabajando con luz ultravioleta.
- **Sensores de movimiento**: constituyen más una medida antirrobo que antimanipulación. Sólo son útiles en un CPD.
- **Membranas conductoras**: forman una malla alrededor del núcleo y son químicamente indistinguibles del encapsulado. Borran las claves en cuanto son expuestas.