Criptografía

Kerberos

PGP

TLS/SSL

SSH

Kerberos

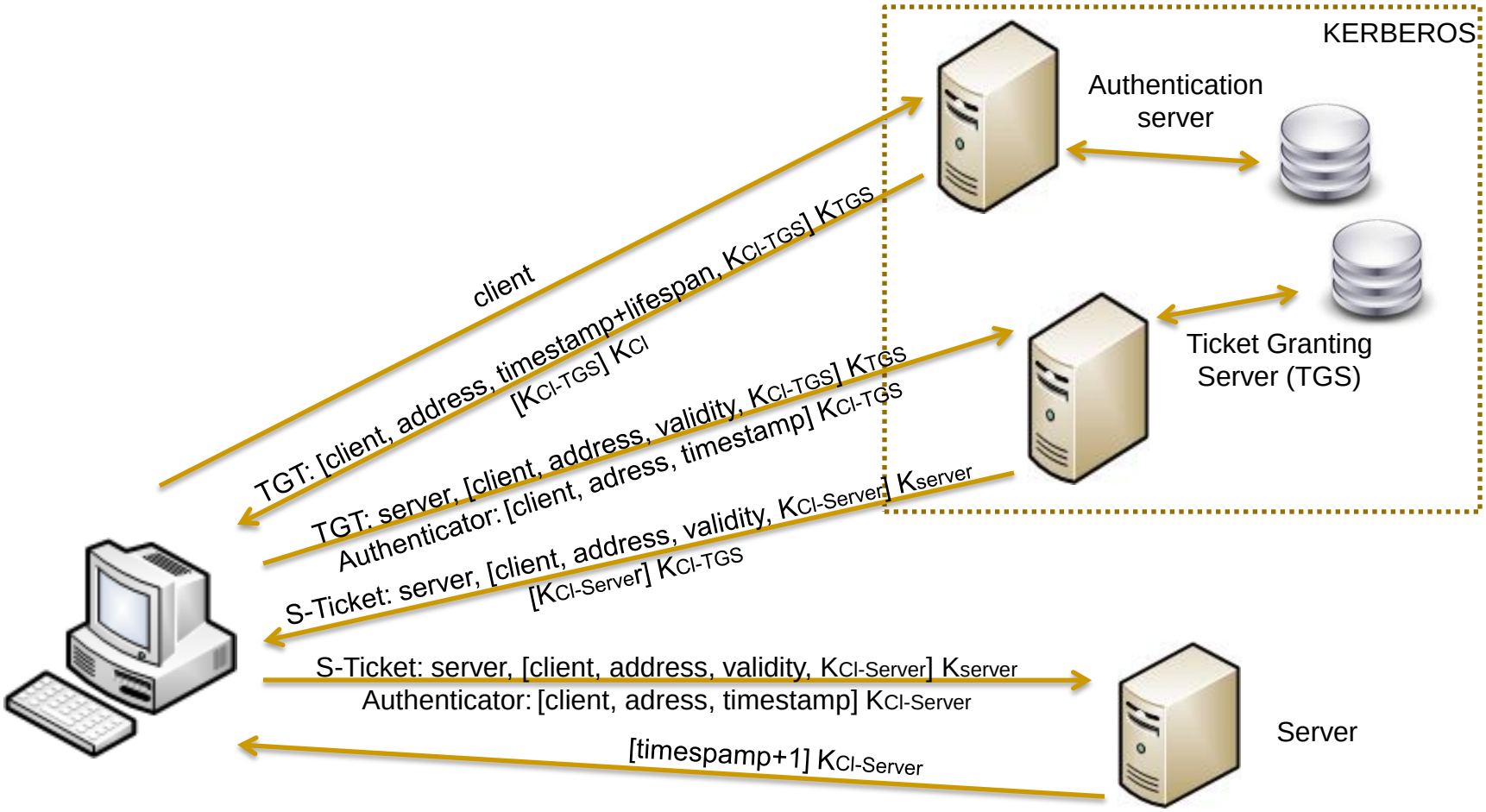
Kerberos - Características

- Protocolo de autenticación.
- Pensado para cliente-servidor. Acceso a servicios distribuidos en una red no segura.
- Provee autenticación mutua.
- Basado en criptografía simétrica.
- Existe una tercera parte de confianza (servidor Kerberos).
- Opcionalmente brinda integridad y confidencialidad de los datos.

Kerberos - Arquitectura

- Basada en:
  - **Clave de sesión**: clave secreta generada por Kerberos y expedida a un cliente para uso con un servidor durante una sesión.
  - **Ticket**: token expedido a un cliente por parte del servicio de tickets de Kerberos para solicitar servicios. Garantiza un cliente autenticado.
  - **Autenticador**: token construido por el cliente y enviado a un servidor para probar su identidad y la actualidad de la comunicación.

Kerberos - Esquema



Kerberos - Desventajas

- **Tickets expiran** a las 8 horas.
- **Centralización**. El servidor principal (Kerberos) debe estar **disponible siempre** y debe **confiarse** en el mismo → se puede utilizar redundancia de servidores
- **Relojes** de servidores deben estar **sincronizados** → se utilizan daemons para sincronizar
- La **seguridad** reside en el **servidor Kerberos**. Si se pudiera acceder al mismo, la seguridad se reduciría a nada.
- **Kerberización**. Si una aplicación quiere hacer uso de Kerberos, debido a su diseño, debe hacer lo que se llama "kerberizarse" lo cual implica tener los fuentes de la aplicación y una gran inversión en tiempo para lograrlo.

Kerberos – Problemas solucionados

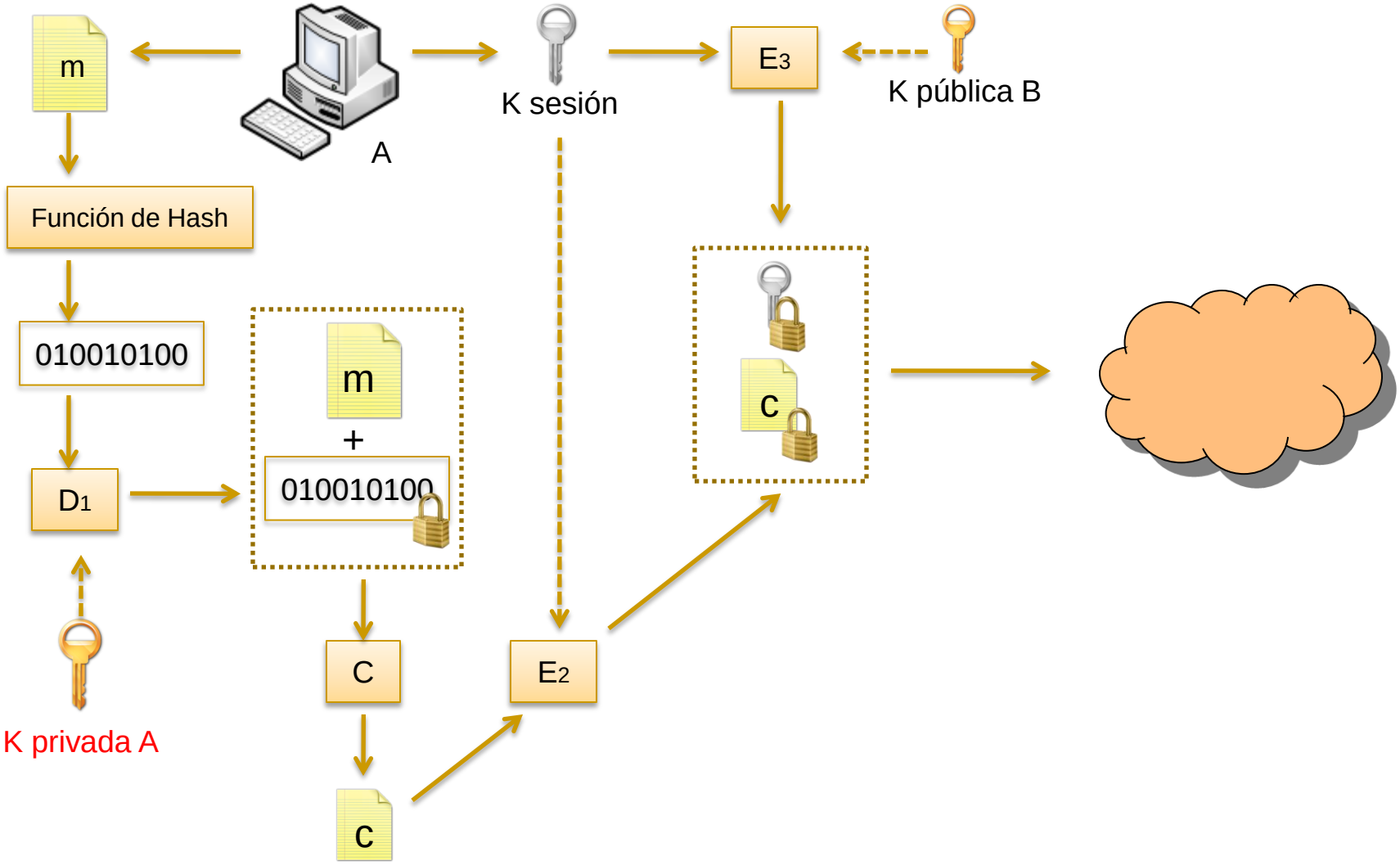
- Cada server de servicios debe conocer todas las claves → únicamente servidor Kerberos las conoce a todas.
- Un ticket nuevo cada vez que se quiere utilizar un servicio → tickets reutilizables.
- Ingresar la clave cada vez que se quiere usar un servicio diferente → separar servidor de autenticación y de emisión de tickets para servicios.
- Autenticación enviando la clave → no se envía la clave en ningún momento. Se utiliza autenticación implícita.
- Reutilizar tickets que queden en la maquina sin estar autenticado (Ataque por REPLAY) → Uso de validity = timestamp + lifespan
- Dentro del lifespan de un ticket, se pueden replicar tickets → Uso de Authenticator
- Posibilidad de capturar tickets + authenticators para replicación → Authenticators no reutilizables (timestamp, lifespan fijo)
- El intermediario se hace pasar por el servidor del servicio solicitado → autenticación mutua (timestamp+1 como response)

PGP (Preety Good Privacy)

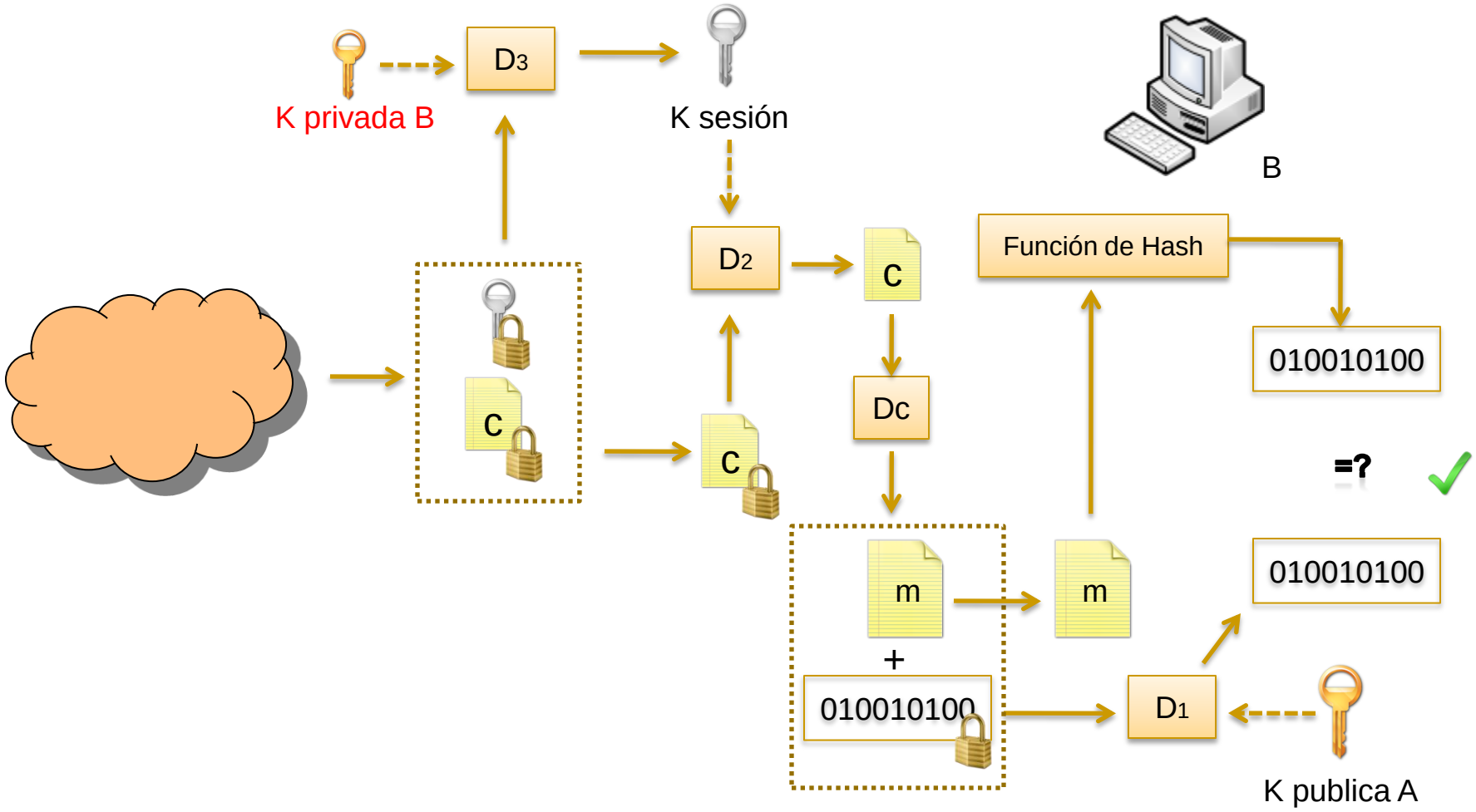
PGP - Características

- Software desarrollado por Phil Zimmermann para brindar confidencialidad, integridad y autenticación a los datos.
- Aplicado especialmente a correo electrónico y archivos.
- Utiliza algoritmos existentes de encripción, tanto simétrica (IDEA, 3DES, etc.) como asimétrica (RSA, DSA, DH), y de dispersión (MD5, SHA-1).
- Soporta compresión de datos (ZIP).

PGP – Esquema (Emisor - A)



PGP – Esquema (Receptor - B)



PGP – Certificación de claves

- PGP reconoce dos tipos de certificado
  - Certificado X.509
    - Requiere Autoridad Certificante.
    - Contiene:
      - Numero de **versión** del estándar.
      - **Clave pública** del portador.
      - **Información** del portador**.**
      - **Numero de serie** del certificado (es un valor único para cada certificado)
      - **Identificador único del portador** del certificado (se le da un identificador según el lugar de procedencia, etc)
      - Periodo de **validez** del certificado
      - **Nombre de la autoridad** que hace el certificado.
      - **Firma** digital de la **autoridad** certificante
      - **Algoritmos de hash y encriptación** utilizados para la firma.
  - Certificado PGP

PGP – Certificación de claves

- Certificado PGP
  - Contiene:
    - Numero de **versión PGP**
    - **Clave publica** del portador. Junto con sus **algoritmo soportado** (RSA, DH (Diffie-Hellman), or DSA (Digital Signature Algorithm))
    - **Información** del portador del certificado
    - La **firma del dueño** del certificado
    - Periodo de **validez** del certificado
    - **Algoritmo preferido** para **encriptado simétrico** (CAST, IDEA o Triple-DES)
  - Pueden contener múltiples firmas (muchas personas pueden firmarlo).
  - Existen servidores adonde subir los certificados para publicar una clave.
  - Su validez se basa en una red de confianza.

PGP – Certificación de claves

- Red de confianza
  - Para validar una clave publica necesito que alguien de fe de su validez. Esa persona se llama **introductor de confianza**.
  - Cualquier usuario podría ser una “autoridad certificante”.
  - Si reconozco a alguien como introductor de confianza, los certificados que firme serán validos para mi.
  - En los “llaveros” de claves publicas se indica el nivel de validez y de confianza para con un usuario y su clave pública.

PGP – Certificación de claves

- Niveles de confianza y validez
  - Nivel de confianza mas alto es el de **confianza implícita** (el que se tiene a la clave propia). Los certificados firmados por esta clave, son validos.
  - Existen tres niveles mas de confianza
    - Confianza completa
    - Confianza parcial
    - Desconfianza
  - Existen, además, tres niveles de validez
    - Valida
    - Parcialmente valida
    - Invalida

TLS (Transport Layer Security)/SSL (Socket Security Layer)

TLS/SSL – Características

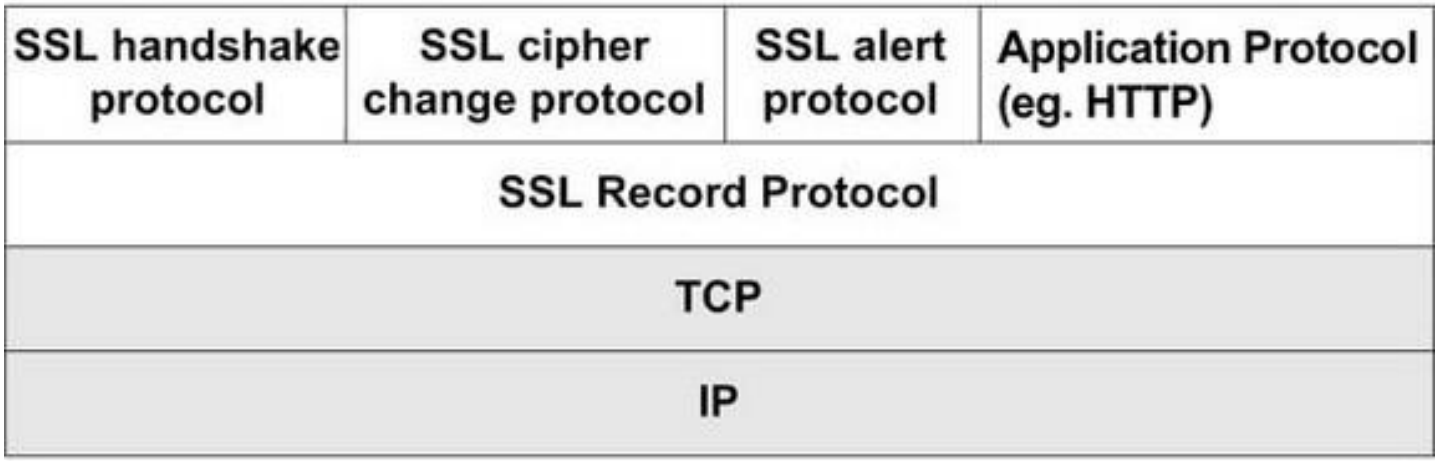
- Protocolo desarrollado para garantizar seguridad en los datos a través de aplicaciones HTTP, SMTP, POP3, etc.
- Trabaja sobre TCP/IP
- Provee:
  - Autenticación: utilizando certificados digitales
    - Puede ser solamente del servidor, del cliente, ambos o ninguno.
  - Integridad: utilizando resúmenes (MAC)
  - Confidencialidad: utilizando encriptado simétrico (DES, 3DES, etc)

TLS/SSL – Características

- Consta de cuatro protocolos
  - Cipher Change Protocol
  - Alert Protocol
  - Record Protocol: asegura la integridad y confidencialidad
  - HandShake Protocol: asegura la autenticación
- Establece sesiones y conexiones
  - Sesión: asociación entre cliente y servidor. Creada mediante el HandShake protocol. Soporta múltiples conexiones.
  - Conexión: enlace transitorio de comunicación.

TLS/SSL - Características

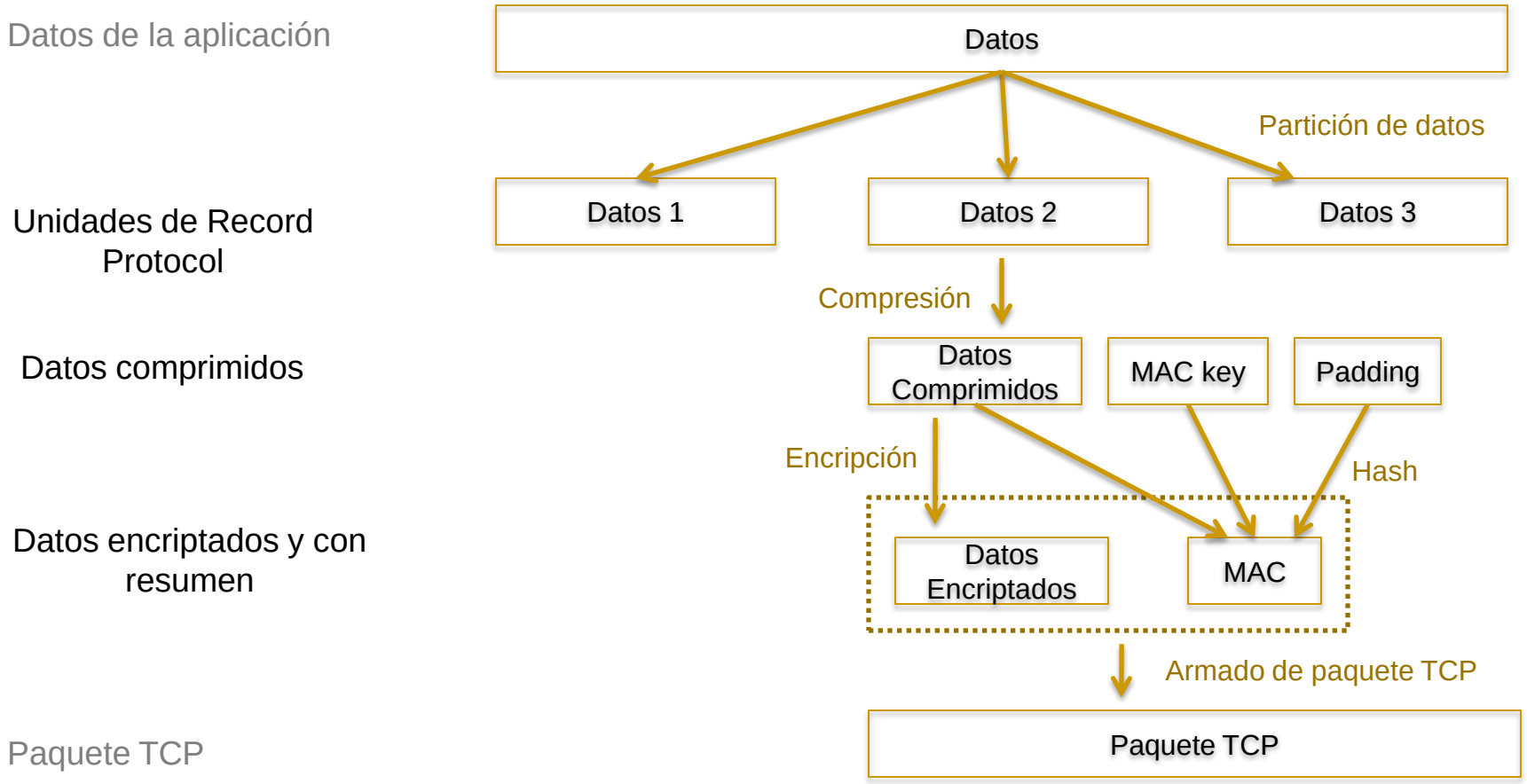
- Arquitectura



TLS/SSL – Protocolos

- Record Protocol
  - Encargado de transferencia de datos.
  - Mensajes y otros protocolos se transmiten mediante este.
  - Fragmenta los datos, y puede comprimirlos para enviarlos.
  - Agrega un **header**, **padding**, y **MAC**, a los datos.
  - El registro se encripta utilizando algún método simétrico.

TLS/SSL – Record Protocol



TLS/SSL – Protocolos

- Alert Protocol
  - Se utiliza para envió de mensajes sobre el funcionamiento del protocolo.
  - Consta de 2 bytes
    - Byte de warning (1), o fatal (2)
    - Byte de código de error
      - Mensaje inesperado
      - MAC incorrecta
      - Falla en el HandShake
      - Certificado incorrecto

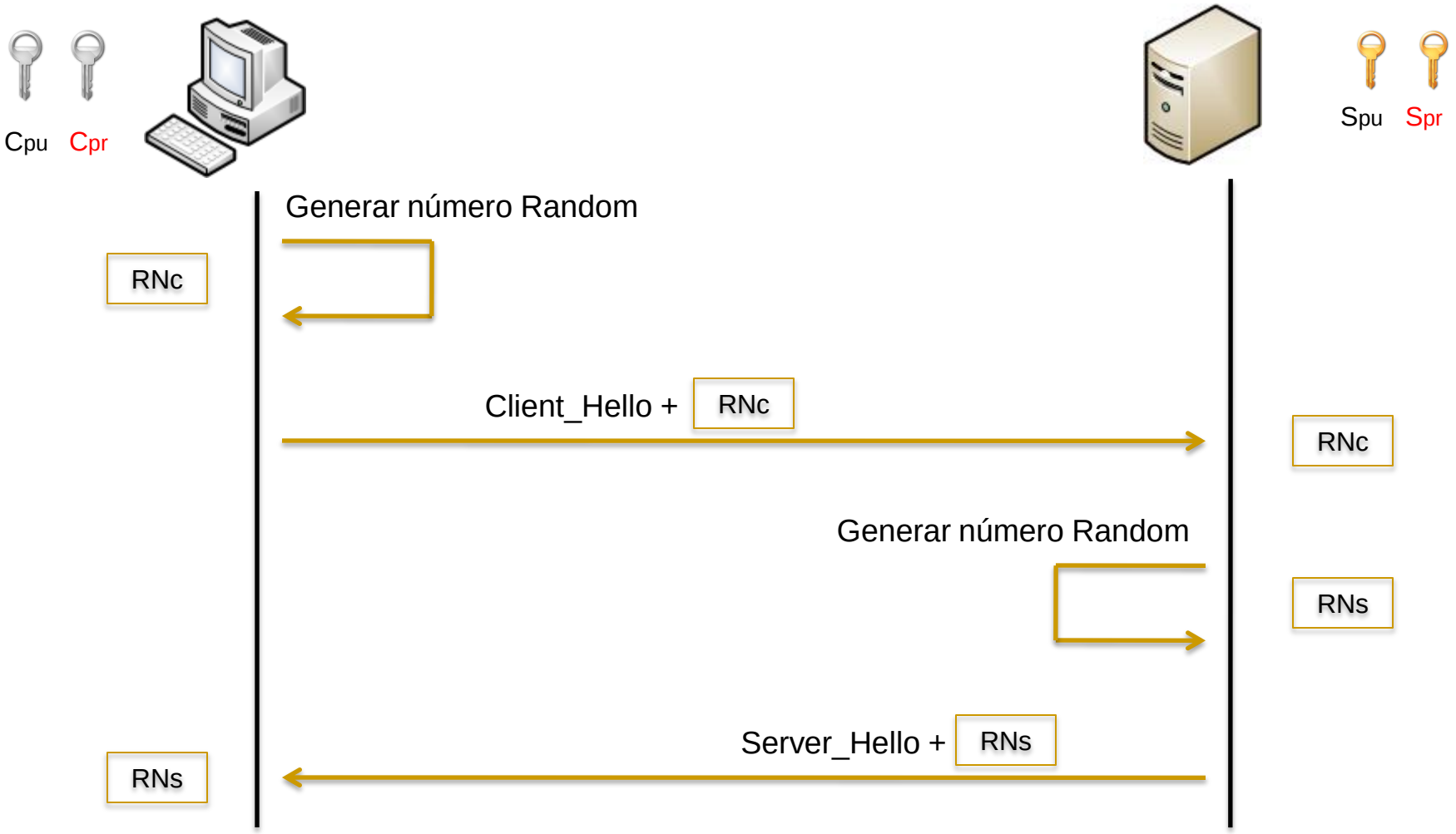
TLS/SSL – Protocolos

- ChangeCipher Spec Protocol
  - Un solo mensaje
  - Lleva el valor (1) por defecto
  - Causa el cambio de estado de escritura/lectura pendiente a actual.

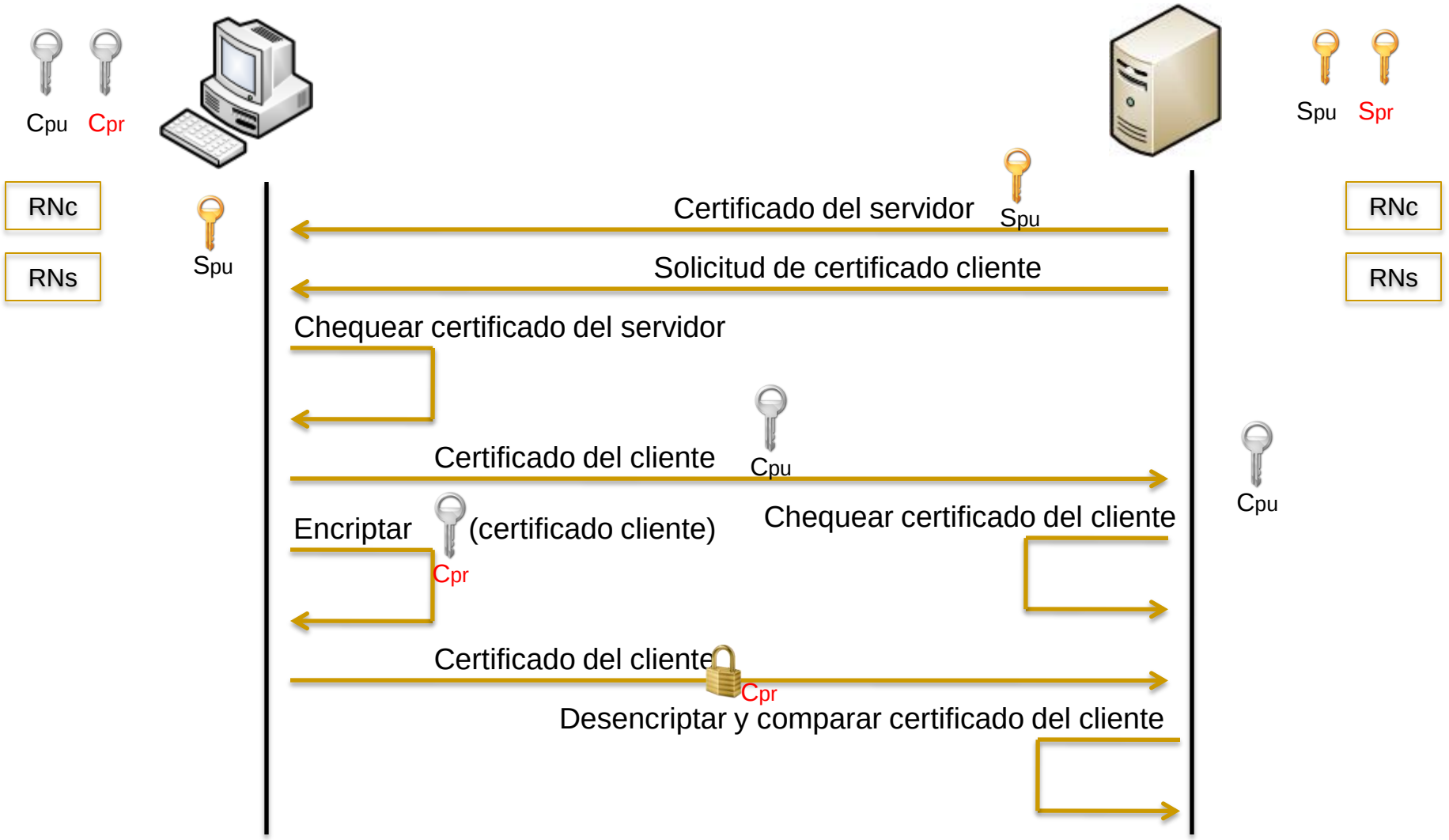
TLS/SSL – Protocolos

- HandShake Protocol
  - Inicia la sesión entre cliente y servidor
  - Se autentican las partes.
  - Se negocia:
    - Métodos de encriptado
    - Métodos de hash para MAC
    - Métodos de compresión
    - Claves para encripción

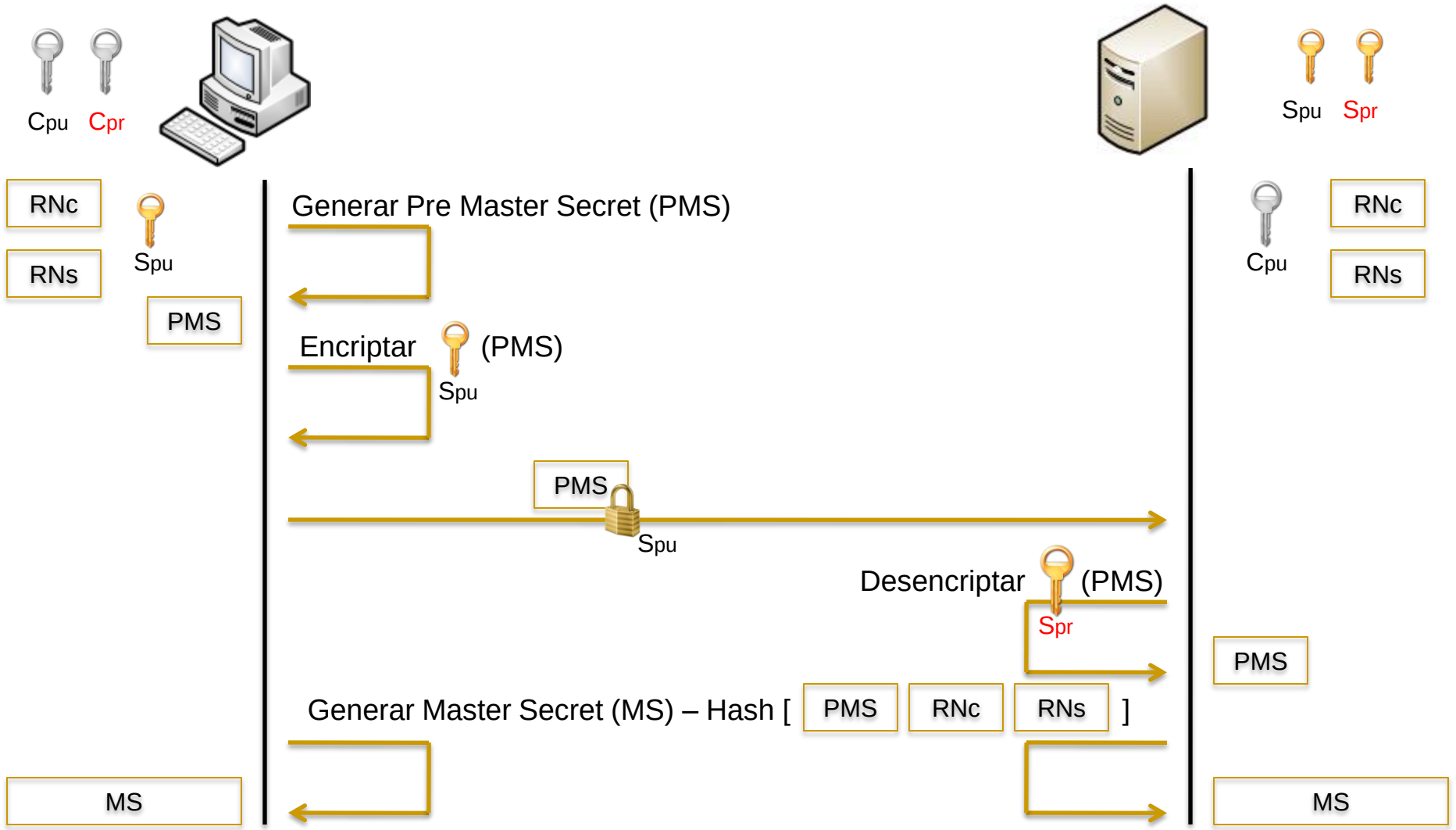
TLS/SSL – HandShake (Fase 1)



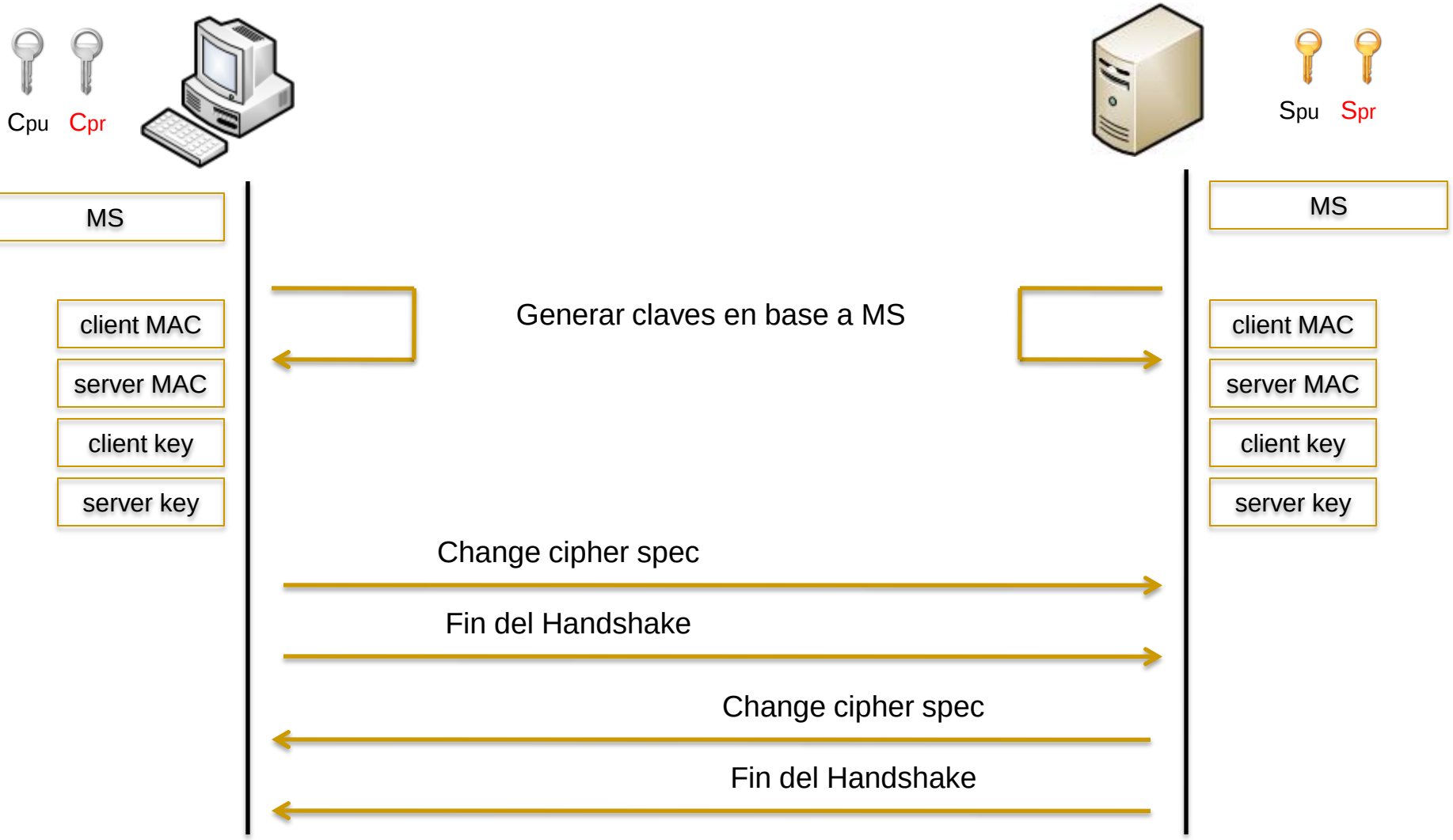
TLS/SSL – HandShake (Fase 2)



TLS/SSL – HandShake (Fase 3)



TLS/SSL – HandShake (Fase 4)



TLS/SSL – TLS

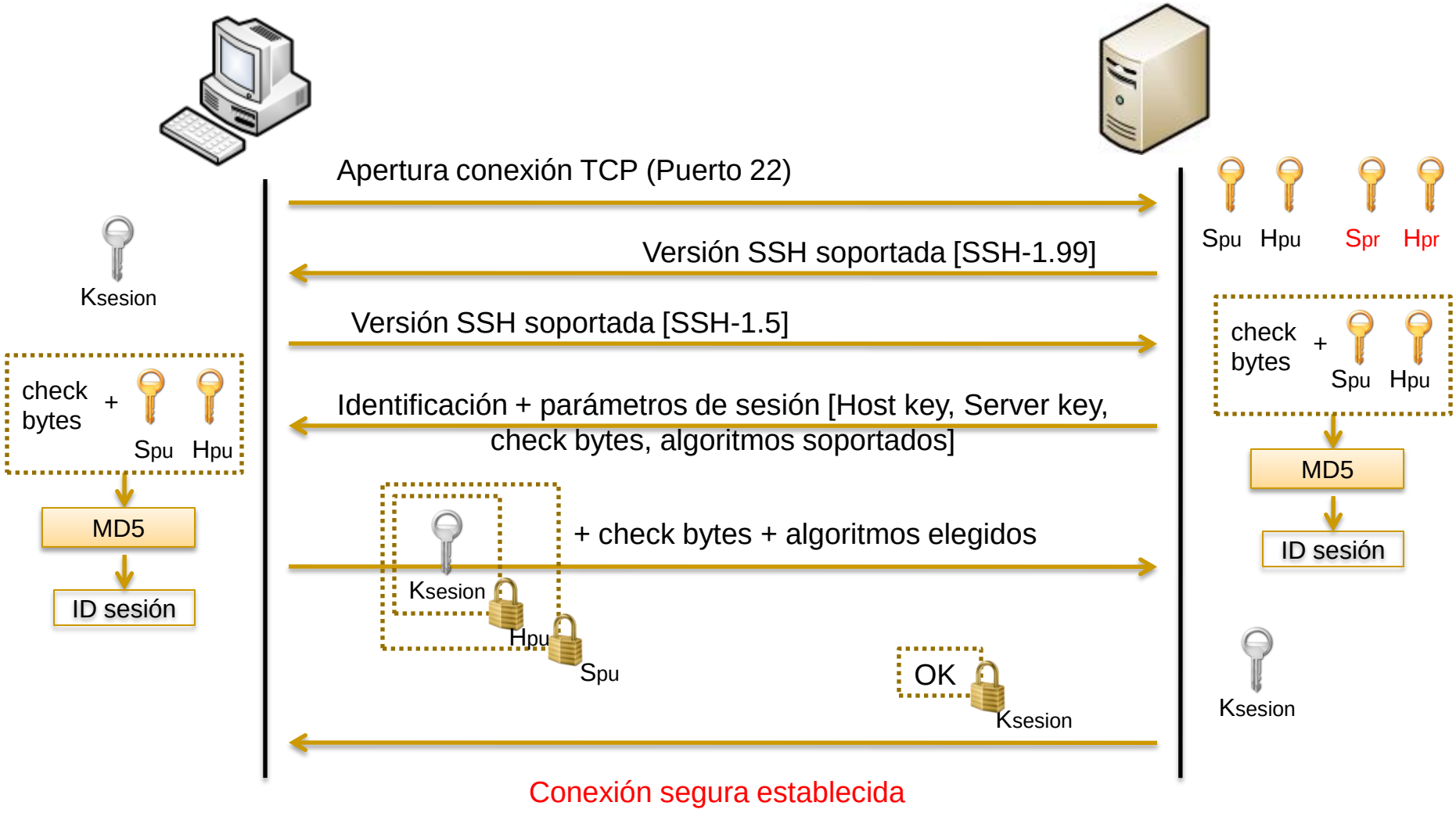
- TLS adopta la versión 3 de SSL como punto de partida.
- Tiene algunas diferencias menores:
  - Formato de numero de versión
  - HMAC en lugar de MAC
  - Se expanden los secret
  - Códigos de alerta adicionales
  - Cambios en métodos de cifrado soportados
  - Cambios en negociaciones de certificados
  - Permite utilizar los puertos estándar

SSH (Secure Shell)

SSH - Características

- Protocolo de red que permite intercambiar datos utilizando un canal seguro entre dos puntos de la red.
- Utilizado mayormente en sistemas Linux y Unix para acceder a un shell remoto de forma segura.
- Diseñado para reemplazar a Telnet por motivos de seguridad.
- Provee confidencialidad, integridad y autenticación.

SSH – Establecimiento de conexión segura



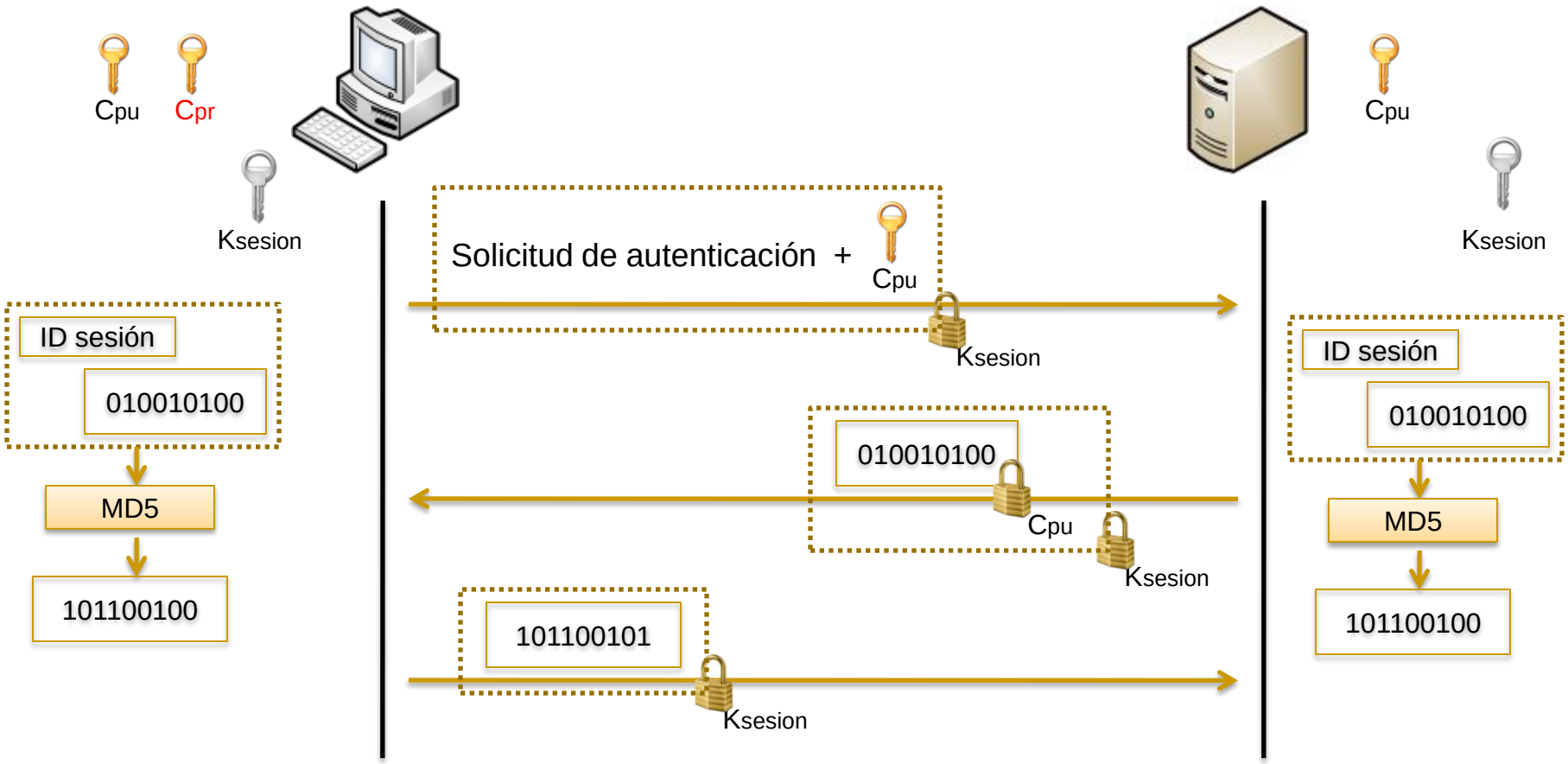
SSH – Autenticación del cliente

- Una vez establecida la conexión, el cliente intentara autenticarse para con el servidor.
- Podría no necesitarse autenticación por parte del cliente.
- Algunos métodos de autenticación soportados son:
  - Password
  - Clave pública
  - Kerberos

SSH – Autenticación del cliente

- Password
  - El cliente envía al servidor su clave secreta, encriptada mediante la clave de sesión. El servidor chequea la validez de la clave para ese cliente en su sistema operativo.
  - Es el método mas simple. No se necesita manipular claves privadas.
  - Es peligroso debido a que la clave es enviada fuera del cliente y podría ser comprometida.

SSH – Autenticación del cliente Clave Pública



SSH – Tecnologías relacionadas

- Kerberos
  - Resuelven problemas similares, pero tienen diferente alcance.
  - Infraestructura diferente.
  - Kerberos esta enfocado a la autenticación.
- PGP
  - Incorporan varios algoritmos de inscripción y hash comunes.
  - PGP se utiliza para encriptar archivos o e-mails.
- TLS/SSL
  - Tienen bastante en común respecto a algoritmos y formas de establecer sesiones.
  - TLS esta enfocado a proveer seguridad al protocolo HTTP.

Bibliografía

- Referencias:
  - Kerberos
    - http://web.mit.edu/kerberos/dialogue.html
    - http://en.wikipedia.org/wiki/Kerberos_(protocol)
  - PGP
    - http://www.pgpi.org/doc/pgpintro/
    - http://www.ietf.org/rfc/rfc4880.txt
  - TLS/SSL
    - http://en.wikipedia.org/wiki/Transport_Layer_Security#TLS_handshake_in_detail
    - http://www.windowsecurity.com/articles/Secure_Socket_Layer.html
  - SSH
    - http://docstore.mik.ua/orelly/networking_2ndEd/ssh
    - http://en.wikipedia.org/wiki/Secure_Shell