Seguridad en Sistemas Distribuidos

30221 - Sistemas Distribuidos

Unai Arronategui, Rafael Tolosana

Dpto. Informática e Ing. de Sistemas

Lectura Recomendada

- Coulouris, Dollimore, Kindberg and Blair, Distributed Systems: Concepts and Design Edn. 5, 2012. Capítulo 11
- Tanenbaum and Van Steen, Distributed Systems: Principles and Paradigms, 3e, (c) 2017 Prentice- Hall. Chapter 9.

Motivación

Motivación

Seguridad

- En los SSDD, existe la necesidad de compartir datos o recursos...

Motivación

Seguridad

- En los SSDD, existe la necesidad de compartir datos o recursos...
- **Acceso seguro** a la información / recursos
  - **Confidencialidad** (*privacy*): protección contra la revelación de información

Motivación

Seguridad

- En los SSDD, existe la necesidad de compartir datos o recursos...
- **Acceso seguro** a la información / recursos
  - **Confidencialidad** (*privacy*): protección contra la revelación de información
  - **integridad**: protección contra la manipulación no autorizada

Motivación

Seguridad

- En los SSDD, existe la necesidad de compartir datos o recursos...
- **Acceso seguro** a la información / recursos
  - **Confidencialidad** (*privacy*): protección contra la revelación de información
  - **integridad**: protección contra la manipulación no autorizada
  - **Disponibilidad**: protección contra una interferencia no deseada

Motivación

Seguridad

- En los SSDD, existe la necesidad de compartir datos o recursos...
- **Acceso seguro** a la información / recursos
  - **Confidencialidad** (*privacy*): protección contra la revelación de información
  - **integridad**: protección contra la manipulación no autorizada
  - **Disponibilidad**: protección contra una interferencia no deseada
- Hay que definir normas de seguridad (*security policies*)
- Las normas utilizarán mecanismos para garantizar la seguridad

Motivación

Norma (policy) vs mecanismo

- *Ejemplo: Seguridad en un edificio*
- Mecanismo 1: reconocimiento facial de alta precisión
- Mecanismo 2: guarda de seguridad

Motivación

Norma (policy) vs mecanismo

- *Ejemplo: Seguridad en un edificio*
- Mecanismo 1: reconocimiento facial de alta precisión
- Mecanismo 2: guarda de seguridad
- Policy 1: el conserje en la entrada del edificio proporciona las credenciales de acceso
- Policy 2 (desastrosa): cualquiera que llegue puede obtener una acreditación en papel sin autorización

Motivación

Norma (policy) vs mecanismo

- *Ejemplo: Seguridad en un edificio*
- Mecanismo 1: reconocimiento facial de alta precisión
- Mecanismo 2: guarda de seguridad
- Policy 1: el conserje en la entrada del edificio proporciona las credenciales de acceso
- Policy 2 (desastrosa): cualquiera que llegue puede obtener una acreditación en papel sin autorización
- una norma desastrosa arruina cualquier mecanismo
  - Policy 2 + Mecanismo 1

Motivación

Amenazas y Ataques

- Objetivo Seguridad:
  - restringir acceso a la información y a recursos, sólo a aquellos autorizados con acceso.
- Las amenazas a la seguridad se dividen en:
  - **Filtración**: adquisición de información por destinatarios no autorizados.
  - **Manipulación**: alteración no autorizada de información.
  - **Vandalismo**: interferencia con el funcionamiento correcto de un sistema sin beneficio para el autor.

Riesgos de Seguridad

- **Espionaje** (*snooping*) [confidencialidad]: Obtención de copias de mensajes sin autorización.
  - Mecanismos: control de acceso, cifrado...

Riesgos de Seguridad

- **Espionaje** (*snooping*) [confidencialidad]: Obtención de copias de mensajes sin autorización.
  - Mecanismos: control de acceso, cifrado...
- **Enmascaramiento** (*fishing*) [integridad]: Enviar o recibir mensajes utilizando la identidad de otro sin su autorización.
  - Mecanismos: cifrado...

Riesgos de Seguridad

- **Manipulación de datos / mensajes** [integridad]: Interceptar mensajes y alterar su contenido antes de transmitirlos al destinatario. El ataque de intermediario es una forma de manipulación de mensajes en la que un atacante intercepta el primer mensaje en un intercambio de claves de cifrado para establecer un canal seguro. El atacante sustituye las claves comprometidas que le permiten descifrar los mensajes posteriores antes de volver a cifrarlos con las claves correctas y transmitirlos.
  - Mecanismos: control de acceso, cifrado...
- **Denegación de servicio** [disponibilidad]: Inundar un canal u otro recurso con mensajes para denegar el acceso a otros.
  - Mecanismos: réplicas...

Mecanismos de Seguridad

Implementan Servicios de Seguridad

- Métodos de control de acceso
- Criptografía
- Limitación de puertos abiertos
- Cortafuegos
- Detección de intrusiones
- Tarjetas de autentificación
- Autentificación biológica

Mecanismos de Seguridad

Implementan Servicios de Seguridad

- Métodos de control de acceso
- Criptografía
- Limitación de puertos abiertos
- Cortafuegos
- Detección de intrusiones
- Tarjetas de autentificación
- Autentificación biológica

Fundamental: nivel de confianza mecanismos y normas (*policies*)

Criptografía

Criptografía

Tipos de Criptografía

- Clave simétrica: misma clave cifrado / descifrado
  - AES, Blowfish...

Criptografía

Tipos de Criptografía

- Clave simétrica: misma clave cifrado / descifrado
  - AES, Blowfish...
- Clave pública: cifrado / descifrado con claves diferentes
  - RSA, DSA, ECC...

Criptografía

Tipos de Criptografía

- Clave simétrica: misma clave cifrado / descifrado
  - AES, Blowfish...
- Clave pública: cifrado / descifrado con claves diferentes
  - RSA, DSA, ECC...
- Funciones hash: a una secuencia de datos (números) se le hace corresponder una secuencia de números (bytes)
  - MD5, SHA1, SHA512...

Criptografía

Tipos de Criptografía

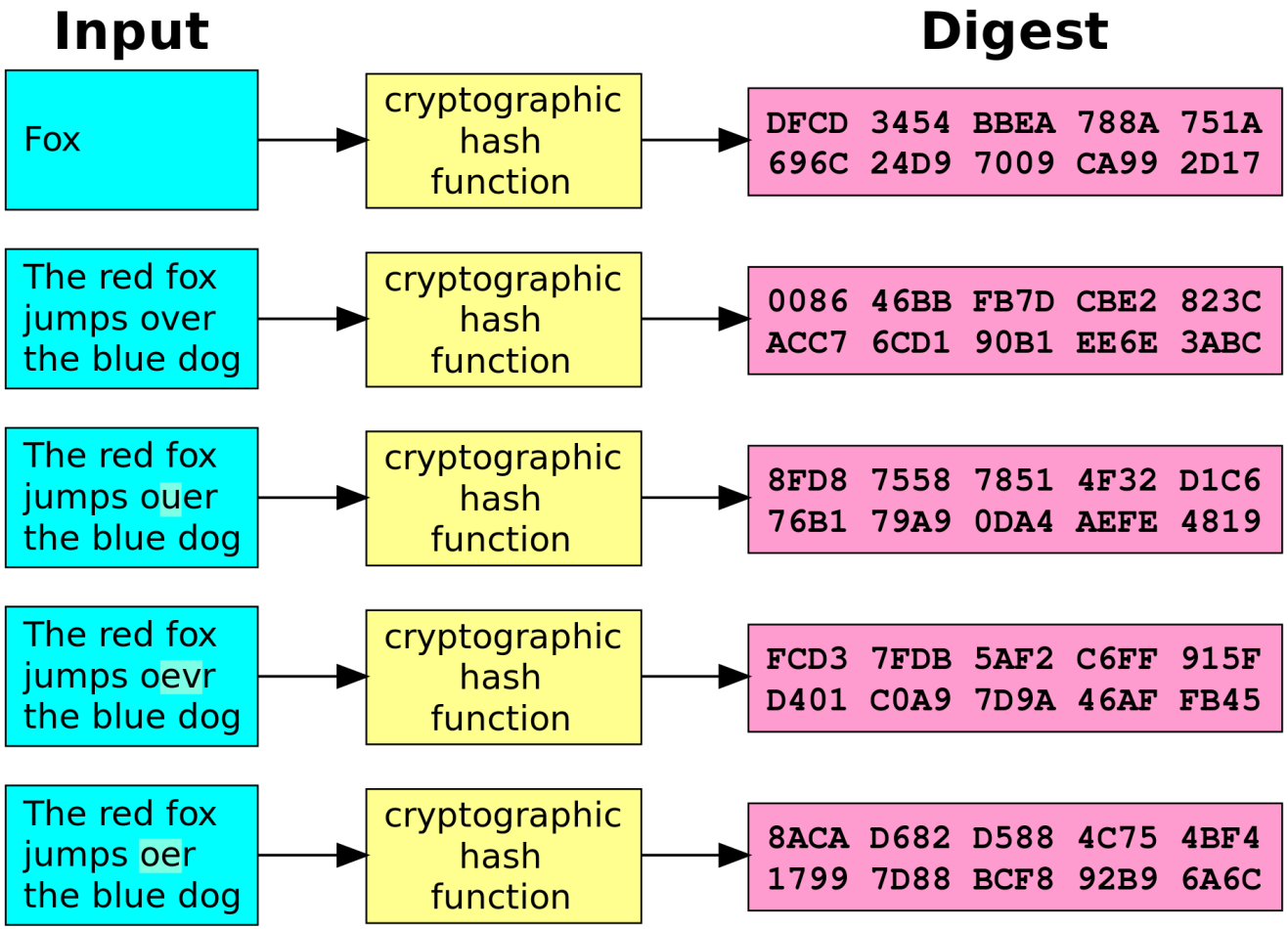
- Clave simétrica: misma clave cifrado / descifrado
  - AES, Blowfish...
- Clave pública: cifrado / descifrado con claves diferentes
  - RSA, DSA, ECC...
- Funciones hash: a una secuencia de datos (números) se le hace corresponder una secuencia de números (bytes)
  - MD5, SHA1, SHA512...

Consideraciones de algoritmos criptográficos:

- Robustez de algoritmo de cifrado y de claves
  - tamaño de claves : RSA, típico de 1024 a 4096; AES, 128, 192 y 256
  - Dificultad computacional de obtención para algoritmos (distribuidos).
- Coste computacional: hash $<$ simétrica $<$ pública

Criptografía

Función Hash Criptográfica



1

[1] fuente: https://en.wikipedia.org/wiki/Cryptographic_hash_function#/media/File : Cryptographic_Hash_Function.svg

Criptografía

Algunas implementaciones de mecanismos de seguridad mediante cifrado para sistemas distribuidos:

- **Nivel aplicación**: Kerberos (simétrico), SSH (combinación)
- **Nivel transporte**: Protocolo SSL/TLS (combinación)
- **Nivel de red**: IPSec (combinación)

Certificados digitales

Mecanismo básico de apoyo para la utilización de cifrado de criptografía asimétrica (de clave pública) en sistemas distribuidos:

- Sirve para **autentificar** una entidad y su información básica en la red e **intercambiar** información de forma segura.

Contenido

  - Clave pública persona / empresa titular del certificado
  - Nombre persona / empresa, fecha caducidad, dirección...
- Además **firma digital de entidad certificadora externa** (tercera)

Certificados digitales

Técnica básica de autentificación del certificado

- firma digital = cifrar(Hash(datos-certificado), Kpriv-CA)

Certificados digitales

Técnica básica de autentificación del certificado

- firma digital = cifrar(Hash(datos-certificado), Kpriv-CA)
- Con la Kpublica CA se puede descifrar el mensaje y comprobar hash(datos-cert)

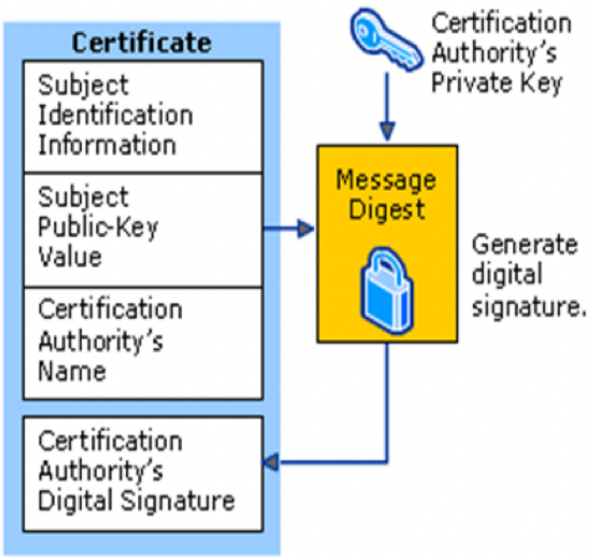
Certificados digitales

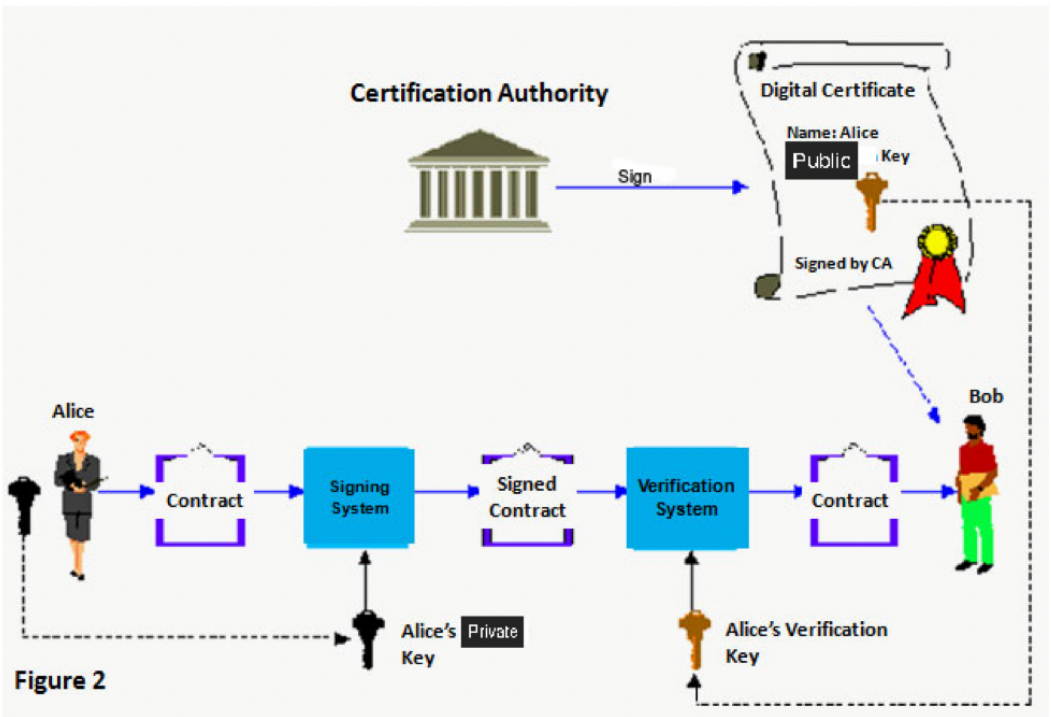
Técnica básica de autentificación del certificado

- firma digital = cifrar(Hash(datos-certificado), Kpriv-CA)
- Con la Kpublica CA se puede descifrar el mensaje y comprobar hash(datos-cert)
- Si la CA se compromete y con ella su Kprivada, hay que anular certificados
  - Protocolo para gestión de revocaciones de certificados

Certificados digitales

Ejemplo de uso del certificado digital





Certificados digitales

Standard más utilizado: certificados digitales X.509

- Formatos de ficheros estandarizados (ejemplo : PKCS 12) incluyen clave privada, certificados X.509 y CRLs (certificate revocaion list).

Posibilidad de auto-firmarse certificados si puedes distribuirlo de forma fiable.

- openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout mikey.pem -out micert.pem

Existe un conjunto de autoridades certificadoras para firmar certificados que puedan utilizarse en Internet (VeriSign, GlobalSign, cacert.org,...)

Certificados digitales

Utilización de este mecanismo y las técnicas de cifrado para proveer diferentes servicios de seguridad en los diferentes entornos Cloud

- En kubernetes en la interacción de los diferentes componentes y los usuarios con esta plataforma:
  - https://kubernetes.io/docs/concepts/cluster-administration/certificates/
  - https://kubernetes.io/docs/concepts/cluster-administration/cluster-administration-overview/securing-a-cluster
- Incluidos protocolos como TLS/SSL (HTTPS), IPSec, SSH, etc

Distribución de Claves Públicas

Distribución de Claves Públicas

Las Claves públicas dan flexibilidad, pero y su distribución fiable?

- Kerberos integra ya la distribución claves secretas (las de sesión), pero también almacena claves secretas que pueden ser comprometidas.
- Uso de Infraestructura de Clave Pública (PKI) para distribución
  - Solo para establecer relación entre identidades y claves públicas a través de certificados digitales.
  - Pero, NO contiene claves privadas (cada uno guarda la suya).
- Modelos de PKIs:
  - Centros de distribución: Autoridades Certificadoras
  - Venden certificados (salvo cacert.org)
  - Cadenas de confianza

Distribución de Claves Públicas

¿Cómo obtener certificados válidos?

- El certificado puede ser obtenido de cualquier forma.
- Pero es la firma digital creada por una entidad fiable (autoridad certificadora o cadena de confianza) la que da validez.

Certificado Raiz de CA (certificado autofirmado) donde se obtiene clave pública para comprobar firmas. Proceso inverso de firma digital:

- A = Descifrar(FirmaDigital, Kpub-entidad-certificadora)
- B =Calcular Hash(Datos incluidos en certificado)
- A = B válida certificado

Distribución de Claves Públicas

Cadenas de confianza (validación recursiva):

- Para confiar en el certificado de Alicia firmado por C2, Bob necesita obtener el certificado de C2.
- Bob utiliza la clave pública de C2 para validar el certificado de Alicia.
- El certificado de c2 está firmado por C1.
- Esto lleva a una cadena de vertificados a validar.
- Termina por un certificado autofirmado por una Autoridad de Certificación raíz, en la que Bob confía.

Distribución de Claves Públicas

Duración de validez de certificados

- Todos los certificados tienen un tiempo de vida limitado para disminuir los riesgos de capturas de la clave privada asociada.
- Después de que haya expirado, debe crearse y certificarse una nueva.
- Además, un certificado puede ser revocado.
- Las revocaciones solo son efectivas si los receptores comprueban regularmente el servidor de certificados para obtener nuevas listas de certificados revocados (CRLs).

Distribución de Claves Públicas

Ejemplo creación certificado

- Generar clave privada
  openssl genrsa -out key.pem 2048
- Generar certificado
  openssl req -new -x509 -key key.pem -out cert.pem -days 3650

Protocolo SSL/TLS

Protocolo SSL/TLS

Utilización de claves públicas provee nuevas posibilidades

- Dar a conocer a todo el mundo claves públicas de entidades a autentificar, sin plantear problemas a la seguridad.
- Autentificación del proceso A por parte del proceso B con protocolo de apretón de manos:
  - B envía a A: cifrar(valor aleatorio, Ka-pub).
  - A contesta con envío a B: cifrar(valor aleatorio + 1, Ka-priv).
  - B verifica (valor aleatorio + 1) al descifrar mensaje anterior con Ka-pub.

Protocolo SSL/TLS

Transport Layer Security (TLS) es el protocolo de autentificación e intercambio de claves para conexiones TCP seguras más utilizado (ej. HTTPS)

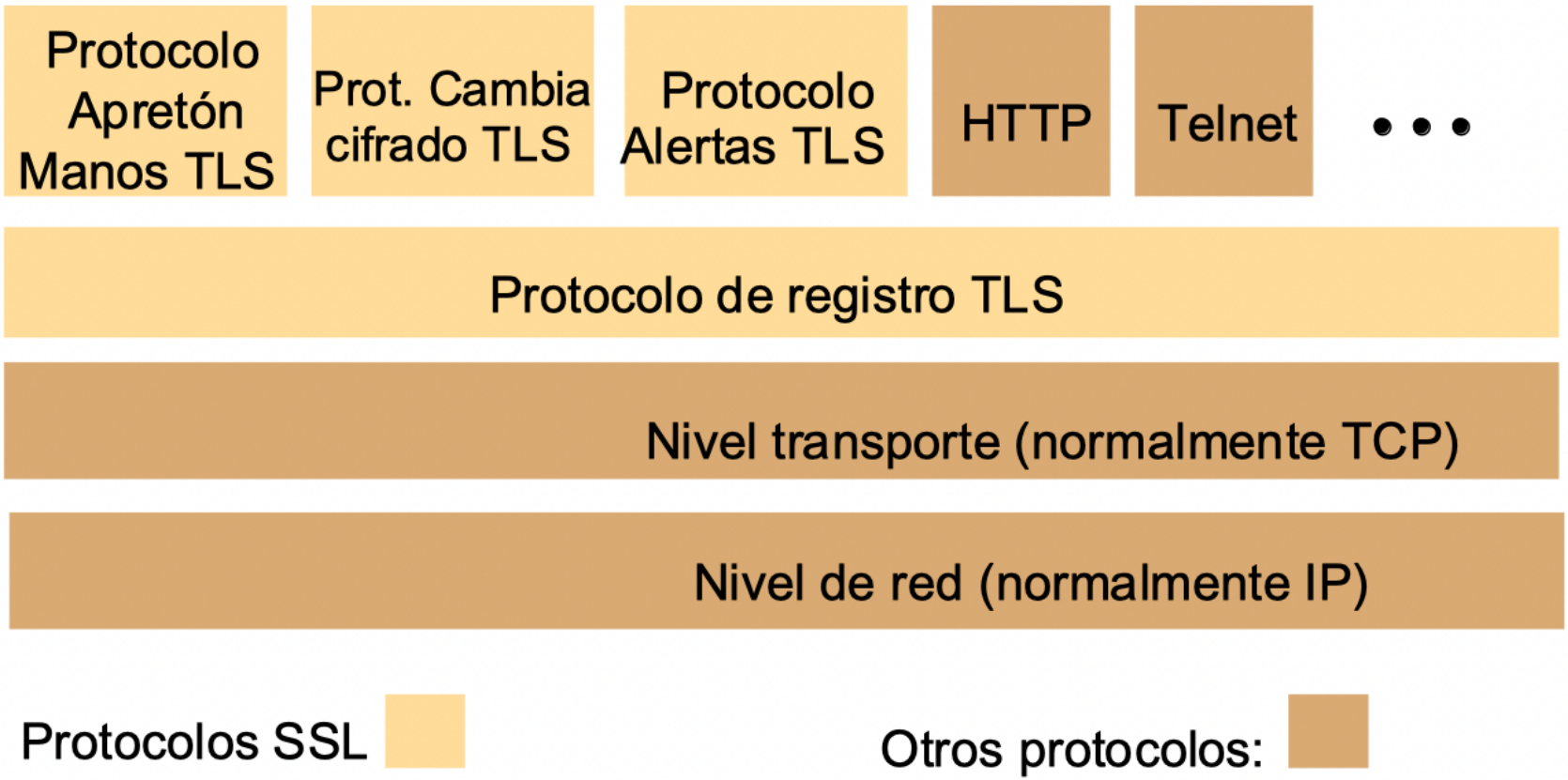
- Inicialmente SSL, convertido en TLS trás estandarizarse (RFC 5246).
- Diseñado para ser extensible.
- Los algoritmos de cifrado son negociados en el protocolo, no son fijos.
- Las aplicaciones lo utilizan como una API parecida a sockets (ej. librería openssl).

Protocolo SSL/TLS

TLS constituido por 3 protocolos de nivel superior

- Protocolo de apretón de manos
  - Negocia algoritmos de seguridad y parámetros
  - Intercambia claves
  - Autentificación de servidor y, opcionalmente de cliente
- Protocolo de cambio de especificación de cifrado
  - Un único mensaje indica final de apretón de manos
- Protolo de alerta
  - Mensajes de error (alertas fatales y avisos)
- Y además el protocolo de registro en el nivel inferior
  - Fragmentación
  - Compresión
  - Autentificación de mensaje y protección de integridad (MAC) Cifrado

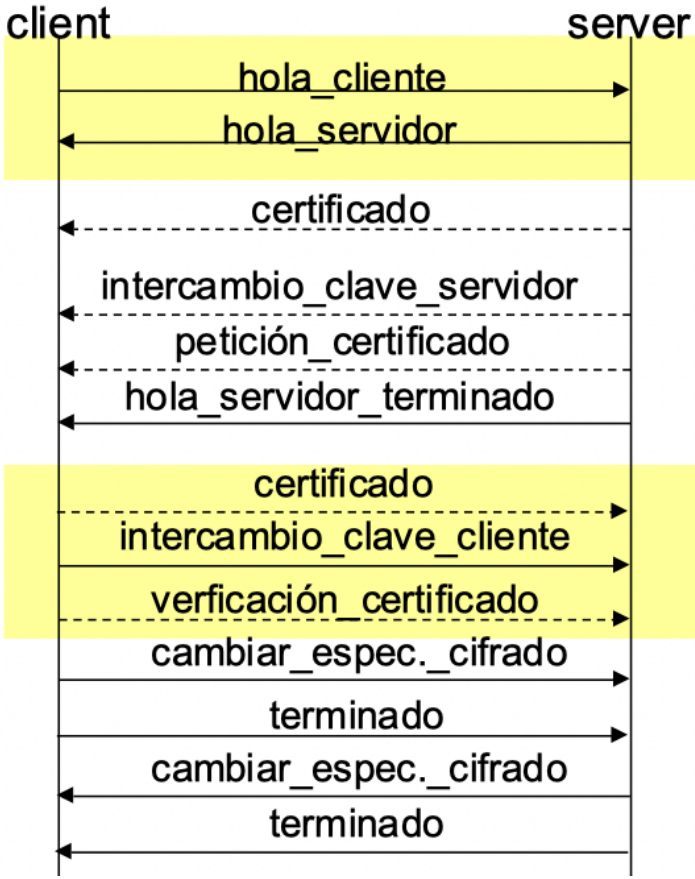
Protocolo SSL/TLS



Protocolo SSL/TLS

Protocolo TLS

Protocolo de apretón de manos en TLS :



Fase 1 : Negociación del ID de sesión, algoritmo de intercambio de claves, algoritmo MAC, algoritmo de cifrado e intercambio inicial de números aleatorios.

Fase 2 : Servidor puede enviar su certificado y mensaje de intercambio de clave, y puede solicitar el certificado del cliente. Notifica final de fase Hola.

Fase 3 : Cliente envía certificado si se lo solicitan y puede enviar un mensaje explicito de verificación de certificado. Cliente siempre envía su mensaje de intercambio de clave.

Fase 4 : Cambia especificación de cifrado y termina apretón de manos.

Kerberos

Kerberos

Problemas de utilización de mecanismos de cifrado

- Coste computacional de algoritmos de cifrado.
- Como distribuir las claves.

Un servicio de autentificación (gestión de identidades) y comunicación segura en sistemas distribuidos: **protocolo Kerberos**

- Autentificación servicios de red, máquinas, usuarios.
- Servicio "Single sign-on" (SSO) para autentificación centralizada. Las contraseñas circulan cifradas a través de la red
- Utiliza solo cifrado simétrico por menor coste computacional (años 198x)....

Kerberos

Problemas de utilización de mecanismos de cifrado

- Coste computacional de algoritmos de cifrado.
- Como distribuir las claves.

Un servicio de autentificación (gestión de identidades) y comunicación segura en sistemas distribuidos: **protocolo Kerberos**

- Autentificación servicios de red, máquinas, usuarios.
- Servicio "Single sign-on" (SSO) para autentificación centralizada. Las contraseñas circulan cifradas a través de la red
- Utiliza solo cifrado simétrico por menor coste computacional (años 198x)....
- Pero mayor dificultad de implementación: compartir claves secretas

Kerberos

Problemas de utilización de mecanismos de cifrado

- Coste computacional de algoritmos de cifrado.
- Como distribuir las claves.

Un servicio de autentificación (gestión de identidades) y comunicación segura en sistemas distribuidos: **protocolo Kerberos**

- Confianza en una entidad tercera donde se depositan claves secretas y permite autentificarse entre entidades que tienen dificultades para intercambiarse claves : Centro de Distribución de Claves (KDC)
- Además de autentificación, (opcionalmente) también puede utilizarse para comunicación confidencial e integra.
- Implementaciones Unix (198x) y Windows (estándar > año 2000)

Kerberos

Planteamiento La entidad tercera (servidor Kerberos, -KDC-) conoce existencia de procesos, máquinas y usuarios

- Conoce una identificación asociada a una clave secreta duradera para cada uno. Cada clave secreta de autentificación es conocida únicamente por KDC y por el elemento (proceso, etc) que se quiere autentificar.
- Posteriormente, KDC provee tickets para servicios de seguridad entre entidades:
  - Autentificados y no falsificables
  - Con mecanismo para impedir ataques de reinyección (replay attacks)
  - Contiene: Identificador, Clave secreta de sesión, tiempo de vida limitado, tiempo creación.
- Utiliza un protocolo para intercambiar tickets de forma segura

Kerberos

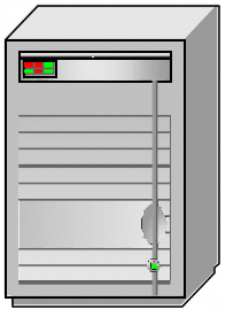
Incluye 3 subservicios de autentificación

- Autentificación inicial de usuario mediante contraseña (clave maestra) frente a KDC:
  - Usuario recibe, de forma segura, ticket de autentificación (ticket granting ticket -TGT-).
    - Contiene clave de sesión de login.
- Solicitar ticket, mediante TGT, del KDC para usar con servidor:
  - Obtener un ticket de servicio (TS) mediante TGT.
- Sesión del servicio: máquina cliente entrega ese ticket TS, sólo válido para una sesión, al servidor.

Kerberos

Protocolo Kerberos

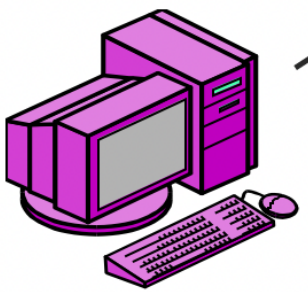
Servidor Kerberos
Centro Distribución Claves
(KDC)



*TGT (Billete para Concesión de billetes) es cifrado para que solo KDC pueda descifrarlo. Contiene información (clave sC,...) que KDC va a leer más tarde.

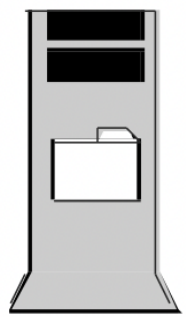
1. Petición de Billete (ticket) para Concesión de billetes (TGT)

2. Respuesta: TGT*; sC** cifrado con clave mC



Cliente (C)

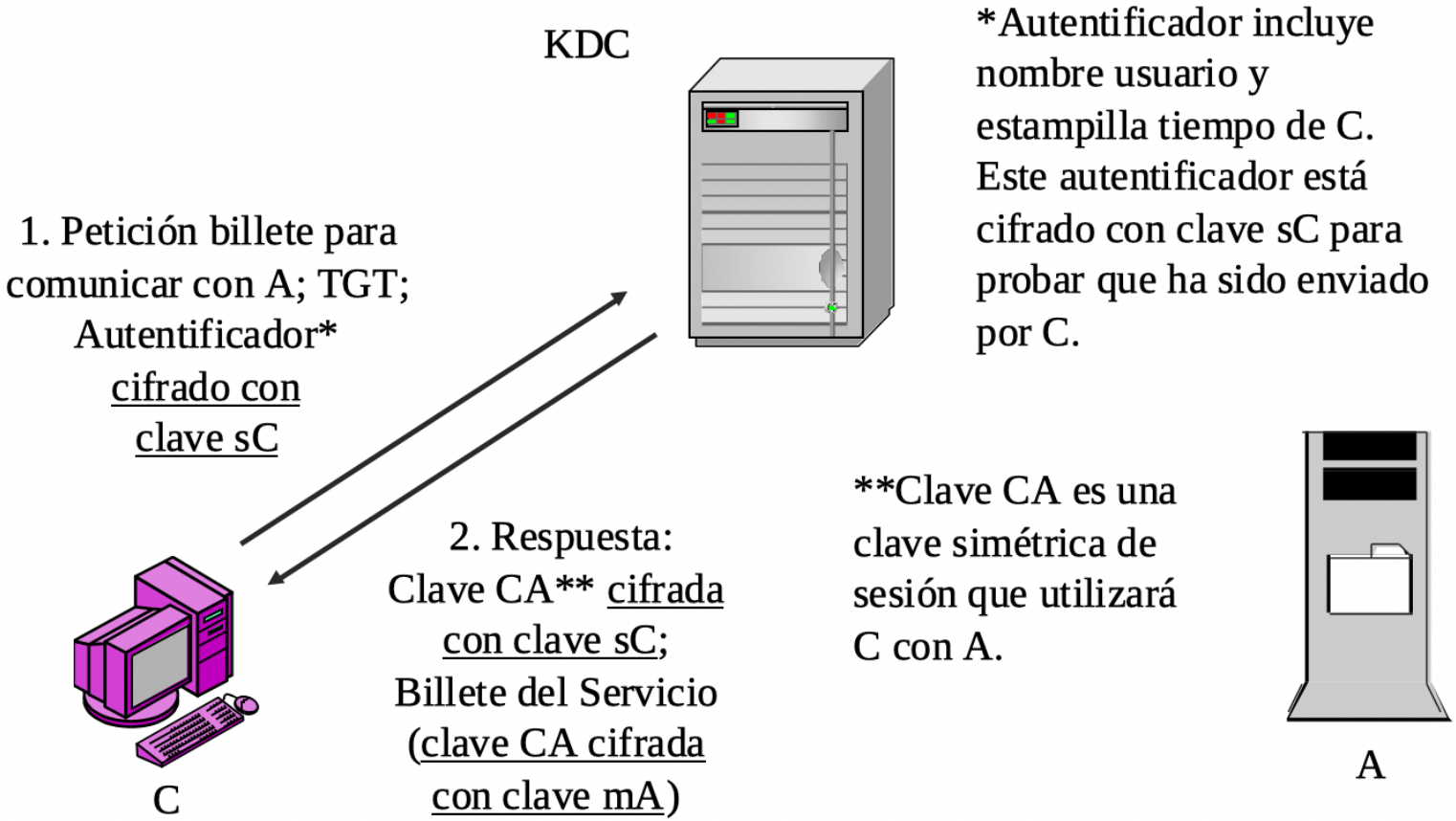
**Clave sC (Clave de entrada en sesión de C) cifrada con la clave maestra de C (Clave mC) ligada a la contraseña de entrada en sesión . En interacciones futuras con KDC, C utilizara sC para limitar exposición de clave maestra.



(A)

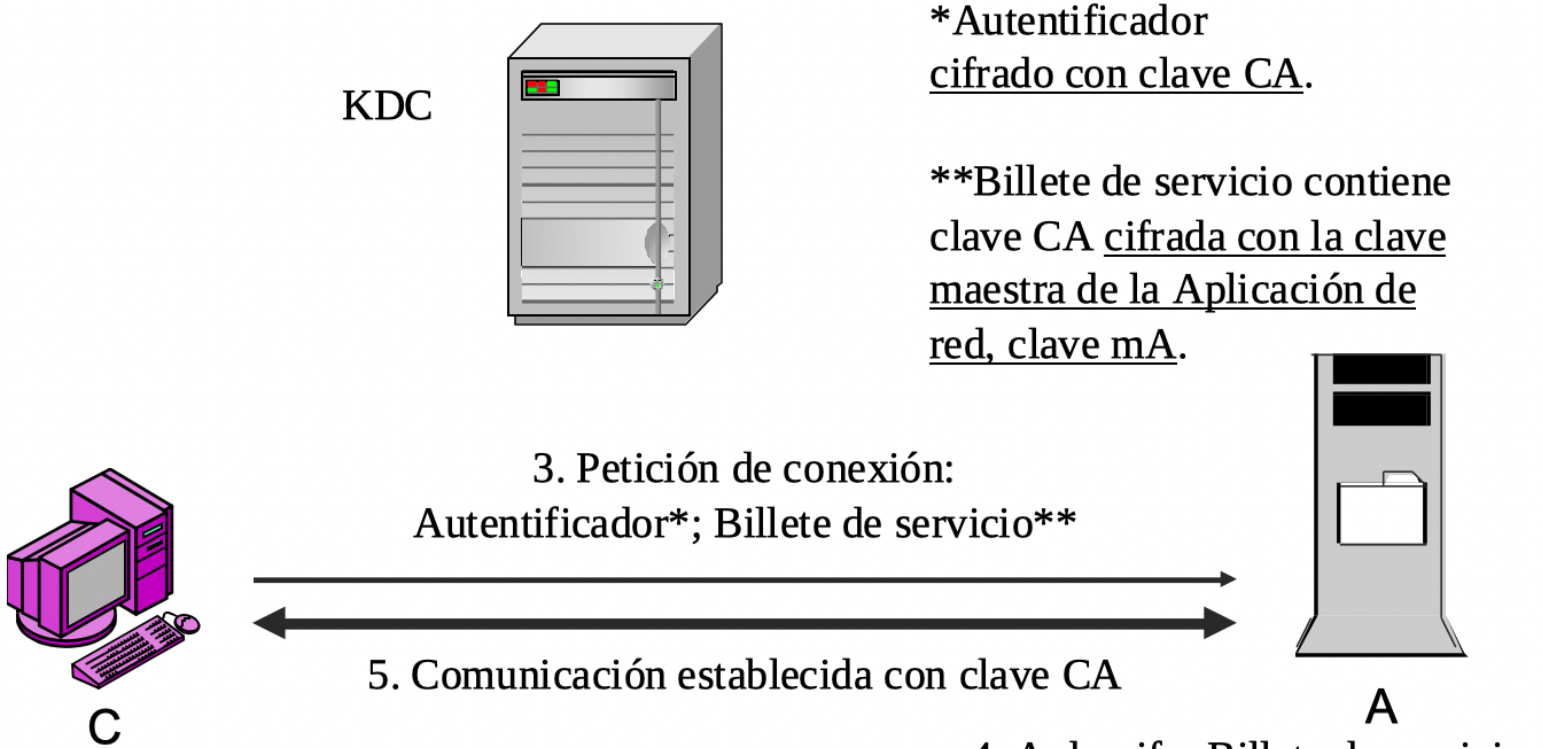
Kerberos

Protocolo Kerberos



Kerberos

Protocolo Kerberos



Kerberos

Debilidad de Kerberos

- Una autoridad central (KDC) que distribuye y contiene todas las claves del sistema distribuido.
- Se utiliza replicación de dicho servicio (KDC), con consistencia eventual, para mitigar el problema de fallo.
- Un acceso no autorizado al servicio central (KDC) compromete las claves.

Seguridad en Sistemas Distribuidos

30221 - Sistemas Distribuidos

Unai Arronategui, Rafael Tolosana

Dpto. Informática e Ing. de Sistemas