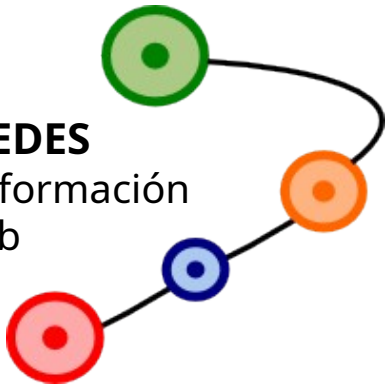
Administración y Gestión de Redes
Lic. en Sistemas de Información

Laboratorio de REDES
Recuperación de Información
y Estudios de la Web



Introducción a la seguridad en Redes de Datos - 3

Equipo docente:

Fernando Lorge (florge@unlu.edu.ar)
Santiago Ricci (sricci@unlu.edu.ar)
Alejandro Iglesias (aaiglesias@unlu.edu.ar)
Mauro Meloni (maurom@unlu.edu.ar)
Patricio Torres (ptorres@unlu.edu.ar)

- 2026 -

Protección de comunicaciones

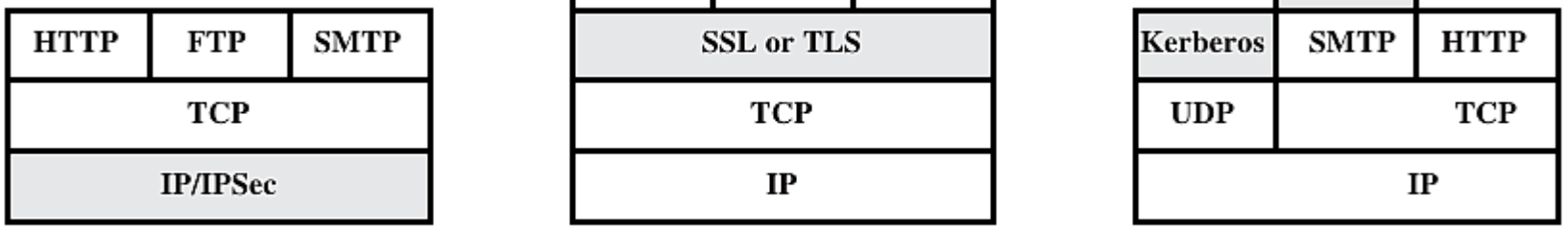
Fantástico, tenemos sistemas criptográficos. Ahora bien, ¿cómo utilizamos estas herramientas para dar **servicios de seguridad** a las comunicaciones?

¿Qué *approach* utilizamos?

Protección de comunicaciones

¿Cómo proteger los datos?

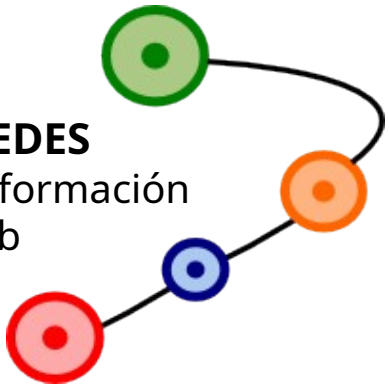
- **Un acercamiento válido es introduciendo protocolos y mecanismos en una o varias capas del modelo OSI**, pudiendo brindar diferentes soluciones a diferentes niveles.
  - End-to-end – Link-level – Network-level - Transport level - Application level
  - PGP, S/MIME, Secure Shell (ssh), Transport Layer Security (TLS), IPSec, L2TP, user-space VPNs



(a) Network level (b) Transport level (c) Application level

Administración y Gestión de Redes
Lic. en Sistemas de Información

Laboratorio de REDES
Recuperación de Información
y Estudios de la Web



OpenPGP

- 2026 -

OpenPGP

Pretty Good Privacy / OpenPGP

- Pretty Good Privacy (PGP) desarrollado por Phil Zimmermann in 1991
- Año 2007: Estándar OpenPGP de IETF (RFC 4880).
- Provee servicios de integridad de datos para mensajes y archivos mediante:
  - Firmas digitales
  - Cifrado (simétrico y de clave pública)
  - Compresión
  - Conversión Radix64
- Además provee administración de claves y certificados.
- Implementación más utilizada: GnuPG

OpenPGP

GnuPG

- Algoritmos soportados:
  - Clave Pública: RSA, ELG, DSA, ECDH, EcDSA, EdDSA
  - Cifrado simétrico: 3DES, CAST5, BLOWFISH, AES, AES192, AES256, TWOFISH, CAMELLIA128, CAMELLIA192, CAMELLIA256, IDEA.
  - Hash: SHA1, RIPEMD160, SHA256, SHA384, SHA512, SHA224
  - Compresión: Sin compresión, ZIP, ZLIB, BZIP2
  - Para envío por SMTP: Radix64, también conocido como “ASCII armor”.
- Alternativa para correo: Secure/Multipurpose Internet Mail Extensions (S/MIME RFC 5751) - Mensaje PKCS#7

OpenPGP

GnuPG

Algoritmos soportados

```
marcelo@marcelo-notebook:~$ gpg --version
gpg (GnuPG) 2.2.4
libgcrypt 1.8.1
Copyright (C) 2017 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <https://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Home: /home/marcelo/.gnupg
Algoritmos disponibles:
Clave pública: RSA, ELG, DSA, ECDH, ECDSA, EDDSA
Cifrado: IDEA, 3DES, CAST5, BLOWFISH, AES, AES192, AES256, TWOFISH,
         CAMELLIA128, CAMELLIA192, CAMELLIA256
Resumen: SHA1, RIPEMD160, SHA256, SHA384, SHA512, SHA224
Compresión: Sin comprimir, ZIP, ZLIB, BZIP2
```

OpenPGP

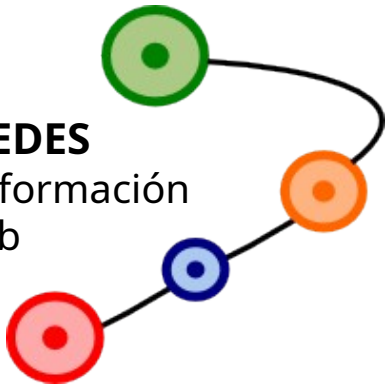
OpenPGP / GnuPG

Casos de uso (ver TP asociado)

1. Cifrar/descifrar un archivo con una clave simétrica
2. Crear par de llaves público/privadas (para cifrado asimétrico)
3. Importar llaves en el llavero
4. Firmar un archivo con mi clave privada
5. Comprobar firma con la clave pública
6. Cifrar un archivo para alguien con su clave pública
7. Descifrar un archivo que recibí con mi clave privada
8. Firmar y Cifrar un archivo
9. Comprobar firma y descifrar archivo

Administración y Gestión de Redes
Lic. en Sistemas de Información

Laboratorio de REDES
Recuperación de Información
y Estudios de la Web



Transport Layer Security (TLS)

- 2026 -

Transport Layer Security

Transport Layer Security (TLS) Protocol

- Basado en Secure Sockets Layer (SSL) desarrollado por Netscape:
  - SSL v2.0: 1995.
  - SSL v3.0: 1996. (RFC Historic 6101, per RFC 7568 -Junio 2015- “SSLv3 MUST NOT be used”)
  - TLS v1.0: 1999 (RFC 2246)
  - TLS v1.1: 2006 (RFC 4346)
  - TLS v1.2: 2008 (RFC 5246)
  - TLS v1.3: 2018 (RFC 8446)

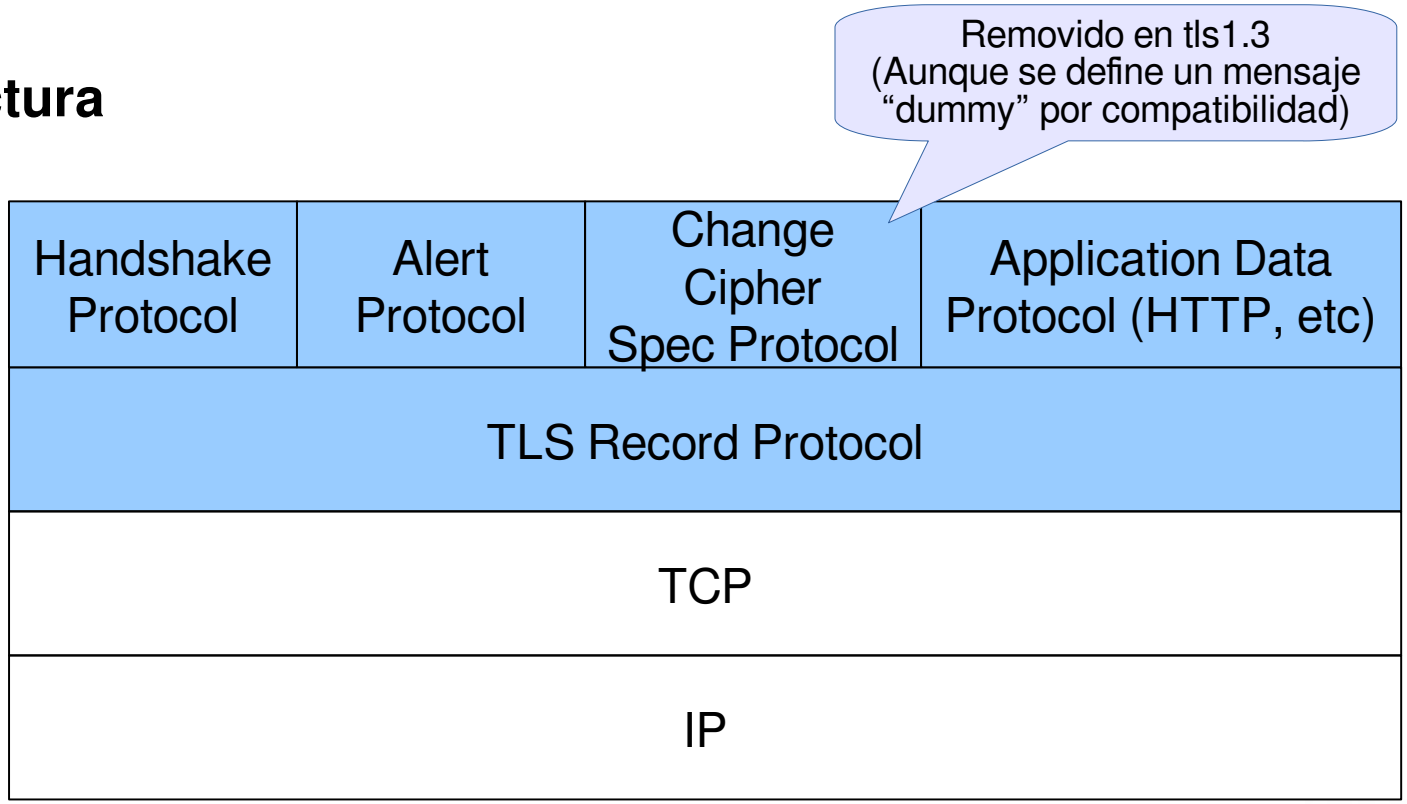
- Implementado sobre TCP proporciona seguridad a protocolos de aplicación como HTTP, SNMP, SIP, etc.

- Provee Cifrado y Autenticación (de 1 o ambos extremos) mediante certificados X.509 (Aunque también es posible utilizar otro tipo de certificados como OpenPGP – RFC 6091)

Transport Layer Security

Transport Layer Security (TLS) Protocol

Arquitectura



El TLS Record Protocol provee confidencialidad e integridad.

Transport Layer Security

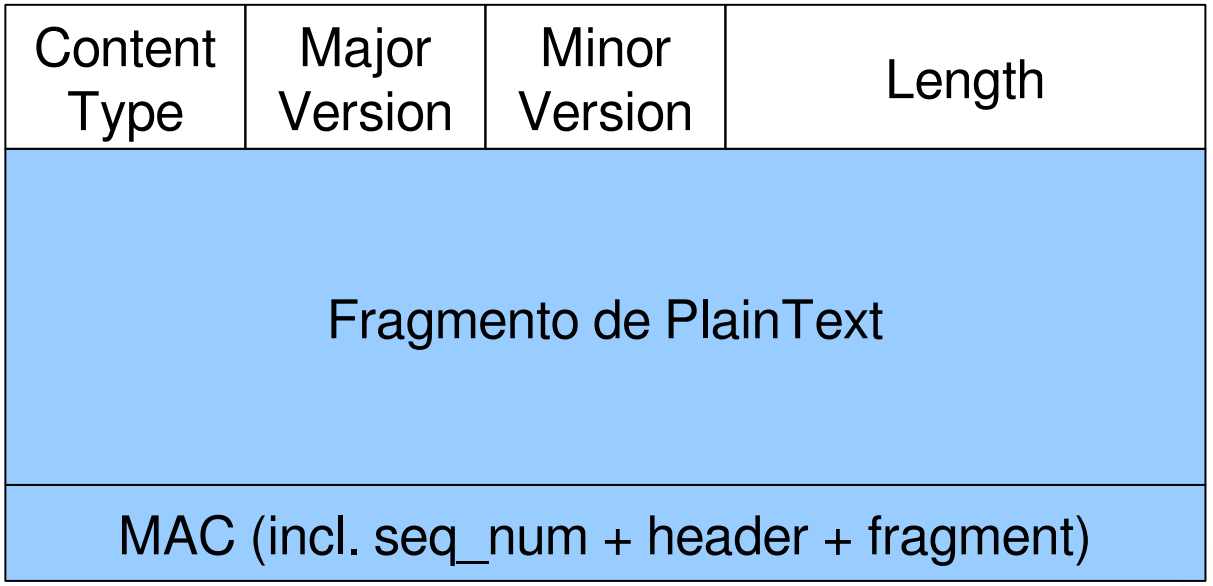
Transport Layer Security (TLS) Protocol
TLS Record Protocol

- **Funcionamiento:**
  - Toma mensajes de aplicación a transmitir (HTTP, SMTP, o los protocolos Handshake, Alert o Change Cipher Spec del propio TLS).
  - Fragmenta y ensambla bloques de 16384 bytes o menor.
  - Solo en versiones < 1.3: Comprime los datos (opcional)
  - Aplica un código de autenticación de mensaje (HMAC definido en RFC 2104)
  - Cifra el mensaje y el MAC calculado utilizando algoritmos simétricos (AES, IDEA, RC2, RC4, 3DES...)
  - Agrega encabezado:
    - Content-Type (protocolo de nivel superior)
    - Major Version
    - Minor Version
    - Compressed Length

Transport Layer Security

Transport Layer Security (TLS) Protocol
TLS Record Protocol V1.2

Formato



Cifrado (Primero se calcula el MAC y luego se cifra)

Transport Layer Security

Transport Layer Security (TLS) Protocol
TLS Record Protocol

- Content Type: Protocolo de nivel superior
  - change_cipher_spec (20)
  - alert (21)
  - handshake (22)
  - application_data (23) (Igual para todos en 1.3)
- Version: Major 3, Minor 3 para TLS v1.2 (Ignorado en v1.3)
- Length: Longitud en bytes del fragmento (No debe superar 2^14 + 2048)
- Fragment: Datos de aplicación protegidos por cifrado +PAD +MAC en v1.2; AEAD para 1.3.
- Ver:
  - https://tools.ietf.org/html/rfc5246#section-6
  - https://tools.ietf.org/html/rfc8446#section-5

Transport Layer Security

Transport Layer Security (TLS) Protocol
Alert Protocol

- Transmisión de mensajes de alerta entre pares
- Mensajes de 2 bytes:
  - El primero indica la criticidad (1-warning o 2-fatal)
  - El segundo indica la alerta específica
- Mensajes de alerta con nivel fatal determina la finalización inmediata de la conexión.
- Cifrado de acuerdo al estado actual.
- Ver:
  - https://tools.ietf.org/html/rfc5246#section-7
  - https://tools.ietf.org/html/rfc8446#section-6

Transport Layer Security

Transport Layer Security (TLS) Protocol
Change Cipher Spec Protocol
(v <1.3)

- Señala el cambio en especificaciones de cifrado y claves negociadas previamente a partir del próximo registro.

- Mensaje único de 1 byte (valor 1) cifrado y comprimido de acuerdo al estado actual.

- Es enviado tanto por el cliente como por el servidor durante el handshake, luego que los parámetros de seguridad hayan sido acordados.

Transport Layer Security

Transport Layer Security (TLS) Protocol
Handshake Protocol

- Permite la autenticación de las partes y la negociación de parámetros de seguridad (algoritmos de cifrado y MAC, claves..).

- Mensajes de 3 campos:

  - Tipo (1 byte)
  - Longitud (3 bytes)
  - Contenido (0+ bytes)

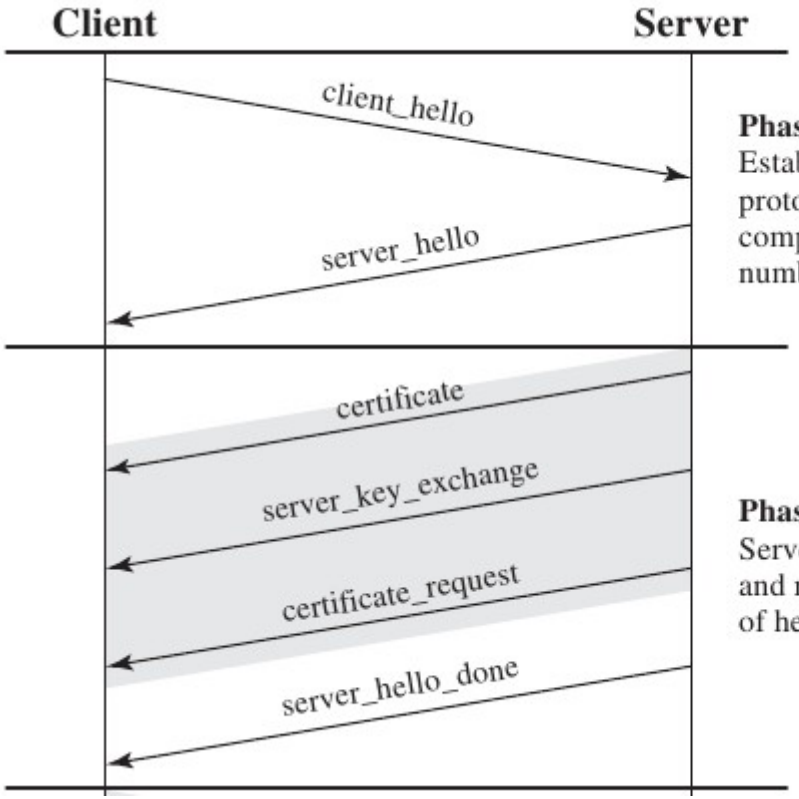
Transport Layer Security

Transport Layer Security (TLS) Protocol
Handshake Protocol (v1.2)

- Se negocia una sesión con los siguientes ítems:
  - Session identifier
  - Peer certificate
  - Compression method
  - Cipher spec (pseudorandom function, bulk data encryption algorithm, MAC algorithm, mac_length)
  - Master secret (48-byte secret compartido entre cliente y servidor)
  - Is resumable (si se permiten nuevas conexiones conservando la sesión)

Transport Layer Security

Transport Layer Security (TLS) Protocol
Handshake Protocol (v1.2)

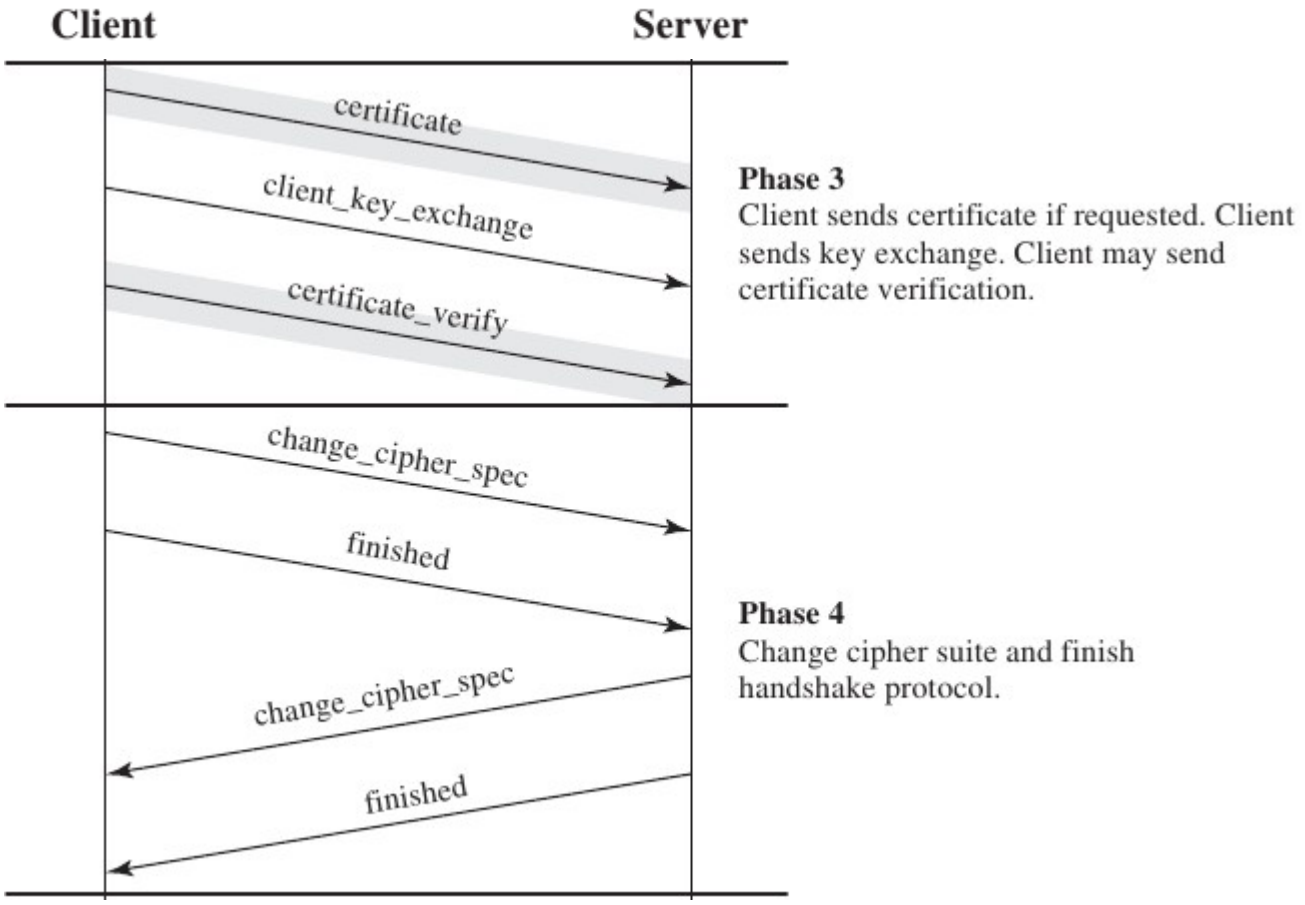


Phase 1
Establish security capabilities, including protocol version, session ID, cipher suite, compression method, and initial random numbers.

Phase 2
Server may send certificate, key exchange, and request certificate. Server signals end of hello message phase.

Transport Layer Security

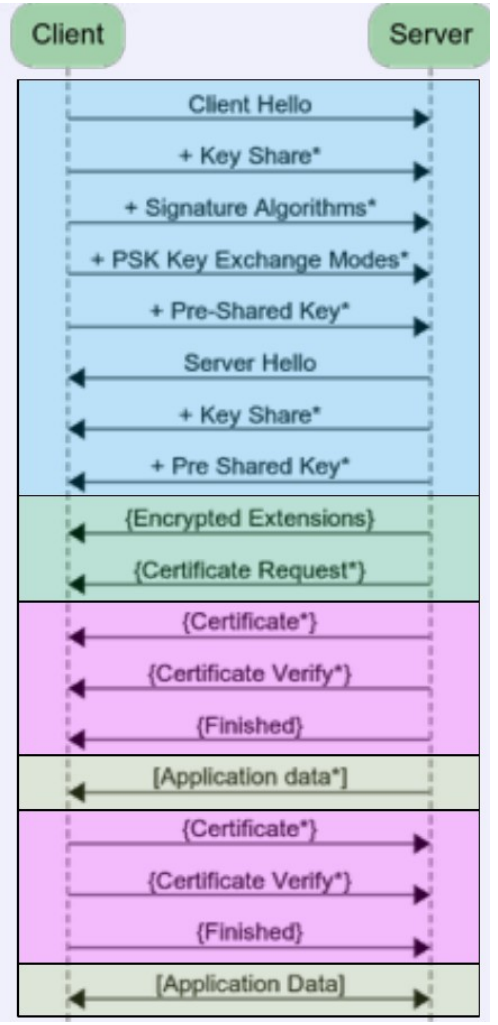
Transport Layer Security (TLS) Protocol
Handshake Protocol (v1.2)



Transport Layer Security

Transport Layer Security (TLS) Protocol
Handshake Protocol (v1.3)

Key Exchange



Server Parameters

Authentication

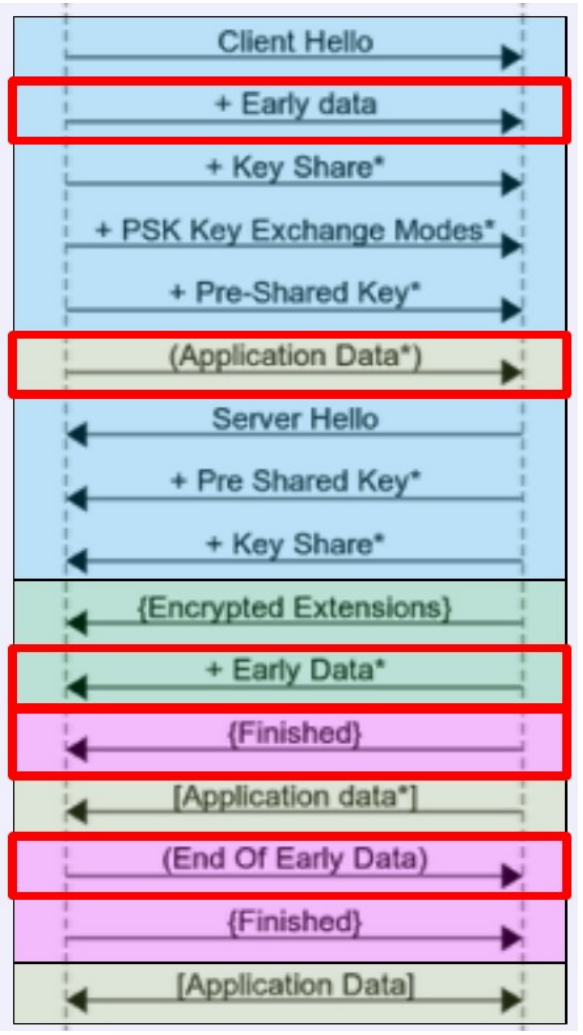
Transport Layer Security

Transport Layer Security (TLS) Protocol

Handshake Protocol (v1.3)

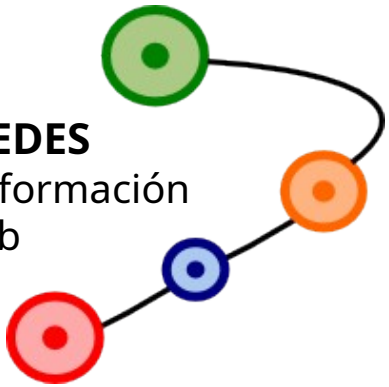
Con pre shared Keys:

Datos en 0-RTT !!



Administración y Gestión de Redes
Lic. en Sistemas de Información

Laboratorio de REDES
Recuperación de Información
y Estudios de la Web



Secure Shell (SSH)

- 2026 -

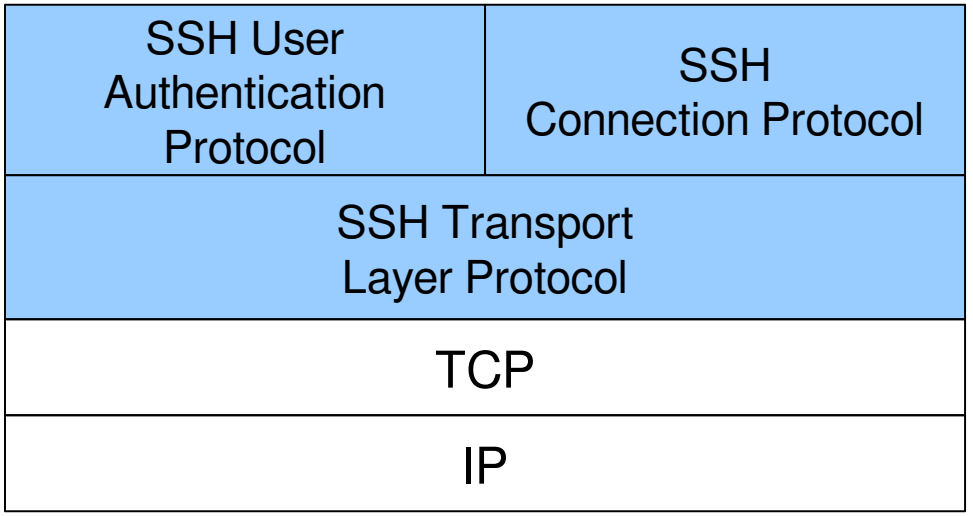
Secure Shell

Secure Shell (SSH)

- Login remoto.
- Tunneling de conexiones TCP/IP.
- Compuesto por
  - Transport Layer Protocol. (RFC 4253)
  - Authentication Protocol. (RFC 4252)
  - Connection Protocol. (RFC 4254)
- Autenticación de hosts mediante “Host Keys”
- Negociación de intercambio de claves, algoritmos de cifrado simétrico y de clave pública, autenticación de mensajes y hash.

Secure Shell

Secure Shell (SSH)



- **SSH Transport Layer Protocol**:

  Provee autenticación, confidencialidad e integridad (opcional compresión)

- **SSH User Authenticacion Protocol**:

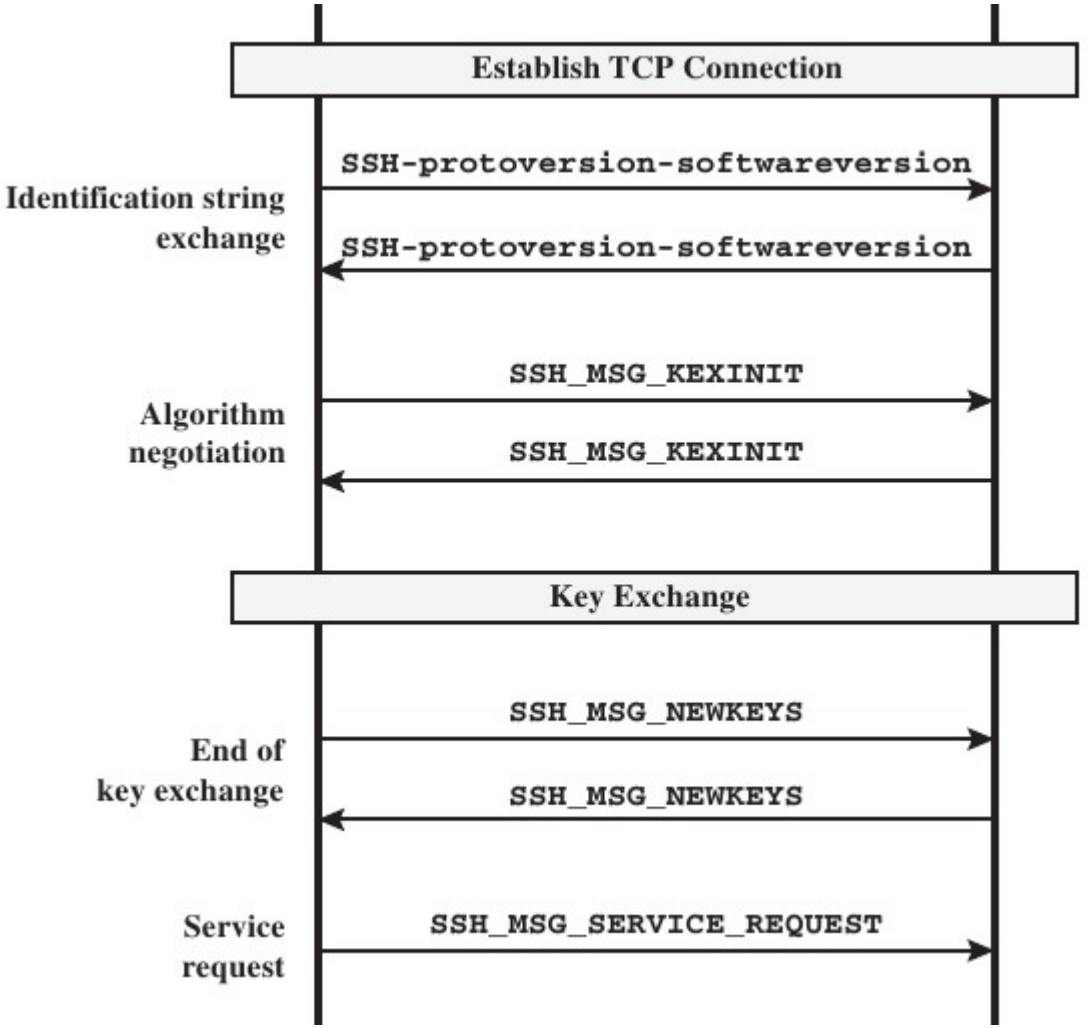
  Autentica usuario frente al servidor

- **SSH Connection Protocol**:

  Multiplexa múltiples canales de comunicación lógicos.

Secure Shell

SSH: Transport Layer Protocol



Secure Shell

SSH: User autenticacion protocol

Métodos de autenticación

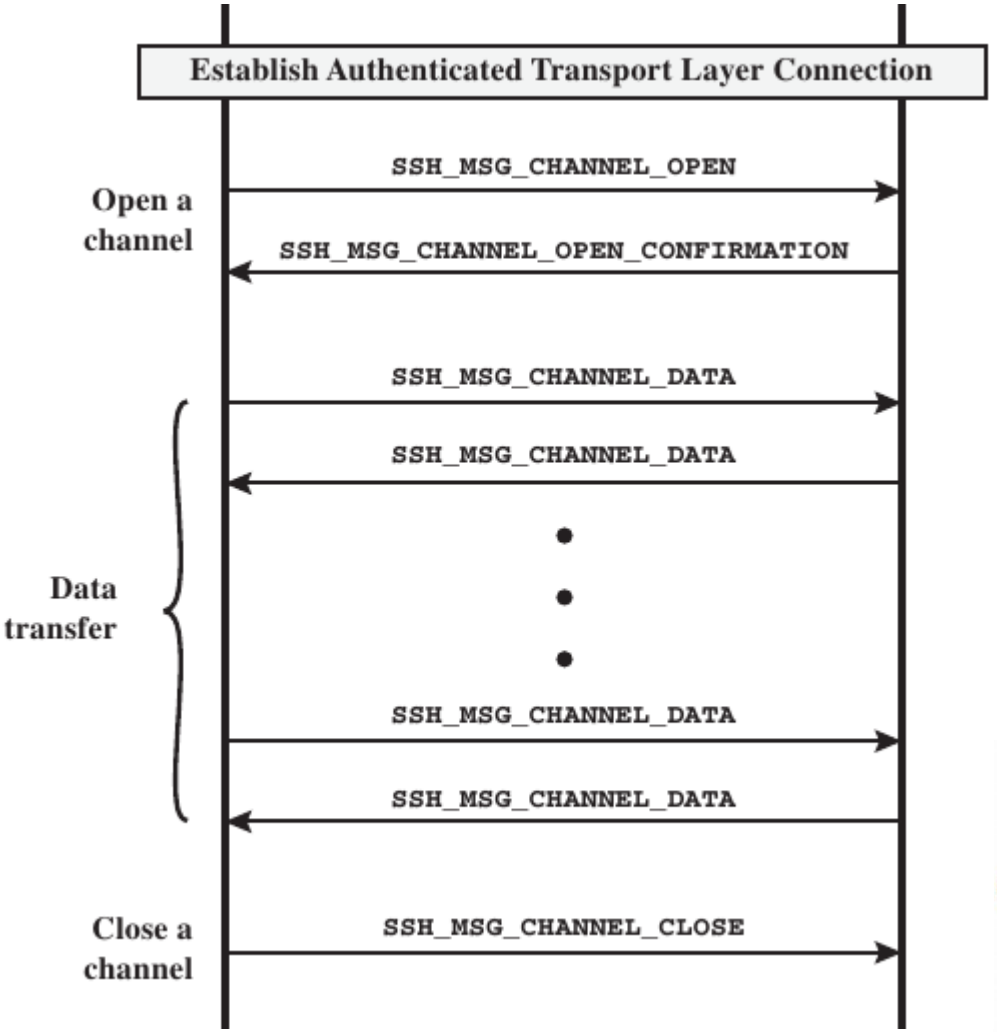
- RFC 4252
  - Clave pública (publickey)
  - Contraseña (password)
  - Basada en host (hostbased)
- RFC 4256
  - Intercambio de mensajes de autenticación genérico (interactivo)

Secure Shell

SSH: Connection Protocol

Tipos de canales:

- Session
- X11
- Direct-tcp
- Forwarded-tcpip



SSH: Connection Protocol

- Direct-tcp:
  - Un puerto en el host local (cliente que inicia la conexión ssh) es redirigido a un host y puerto en el lado remoto.
  - Ejemplo:
    - -L [bind_address:]port:host:hostport
    - # ssh -L 127.0.0.1:80:intra.example.com:80 gw.example.com
- Forwarded-tcpip
  - Un puerto en el host remoto (servidor al que se conecta el cliente ssh) es redirigido a un host y puerto en el lado local.
  - Ejemplo:
    - -R [bind_address:]port:host:hostport
    - ssh -R 8080:localhost:80 public.example.com

Bibliografía

- STALLINGS, W. 2011. *Cryptography and Network Security - Principles and Practice* (5ht ed). Prentice Hall.
  - Capítulo 2: Classical Encryption Techniques
  - Capítulo 3: Block Ciphers and the DES
  - Capítulo 9. Sección 1: Principles of Public-Key Cryptosystems
  - Capítulo 11: Cryptographic Hash Functions
  - Capítulo 12: Message Authentication
  - Capítulo 13: Digital Signatures
  - Capítulo 16: Transport-Level Security
  - Capítulo 18. Sección 1: Pretty Good Privacy (PGP)

Próxima: VPNs