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 - 2

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)
Marcelo Fernandez (fernandezm@unlu.edu.ar)

- 2020 -

Administración y Gestión de Redes
Lic. en Sistemas de Información

Laboratorio de REDES
Recuperación de Información
y Estudios de la Web

Criptografía

Cifrado por Sustitución / Transposición
Criptografía Simétrica / Asimétrica
Esteganografía
Funciones Hash
Código de Autenticación de Mensajes
Firma Digital

- 2020 -

Sistemas Criptográficos
Clasificación I

- Por **tipo de operaciones usadas para transformar** el texto plano en texto cifrado:
  - *Sustitución*: cada elemento del texto plano (bit, letra) se mapea a otro elemento
  - *Transposición*: los elementos del texto plano son reacomodados.

Sistemas Criptográficos
Clasificación II

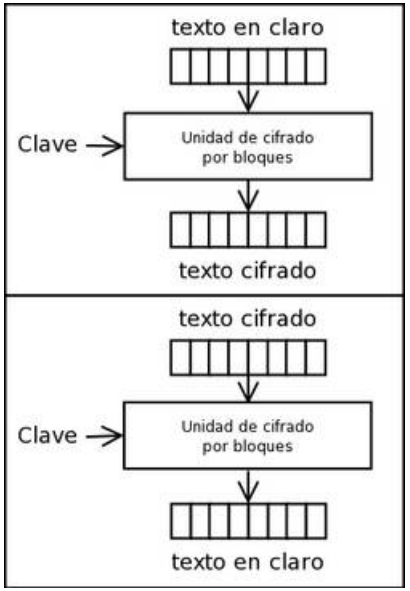
- Por el **número de claves utilizadas**:
  - *Simétricos*: emisor y receptor utilizan la misma clave. También se los suele denominar como de una clave, de clave secreta o cifrado convencional.
  - *Asimétricos*: Emisor y receptor utilizan claves diferentes. Denominados también de doble clave, o cifrado de clave pública.

Sistemas Criptográficos
Clasificación III

- Por la **forma en que el texto plano es procesado**.
  - *Cifrado por bloque*: procesa un bloque de elementos por vez, produciendo un bloque de salida por cada bloque de entrada.
  - *Cifrado continuo (stream)*:Procesa los elementos de manera continua, produciendo un elemento de salida por vez, a medida que se va alimentando.

Criptografía

Sistemas Criptográficos
Clasificación III



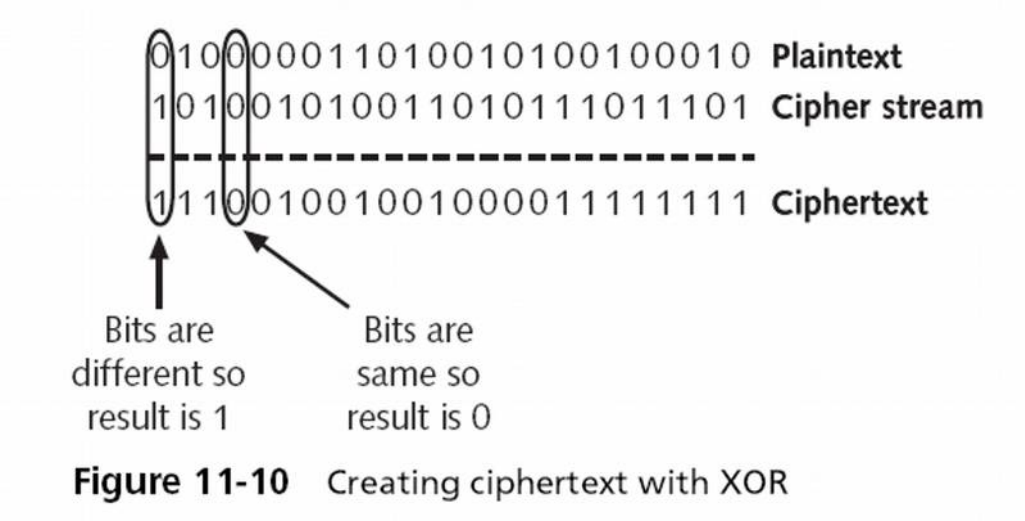


Figure 11-10 Creating ciphertext with XOR

Cifrado por bloques

Cifrado Continuo

Criptosistema

- **m: (Plaintext)** mensaje en claro o grupos de mensajes en claro que se desean cifrar.
- **c: (Ciphertext)** mensaje o grupos de mensajes cifrados
- **K: (Keys)** conjunto de claves que se emplean en el criptosistema
- **E: (Encryption Algorithm)** es el conjunto de transformaciones de cifrado o familia de funciones que se aplica a cada elemento de m para obtener un elemento de c. Existe una transformación diferente, denominada E_k, para cada valor posible de la clave k.
- **D: (Decryption Algorithm)** es el conjunto de transformaciones de descifrado.

Todo criptosistema cumple $D_k(E_k(m)) = m$

Criptografía

Esteganografía

- Estudia los procedimientos encaminados a ocultar la existencia de un mensaje en lugar de ocultar su contenido.
- El objetivo de la esteganografía es ocultar el mensaje dentro de otro sin información importante, de forma que el atacante ni siquiera se entere de la existencia de dicha información oculta.
- No se trata de sustituir al cifrado convencional sino de complementarlo: ocultar un mensaje reduce las posibilidades de que sea descubierto; no obstante, si lo es, el que ese mensaje haya sido cifrado introduce un nivel adicional de seguridad.

¿Se les ocurren ejemplos?

Esteganografía - Ejemplos

- Tinta invisible.
- Marcas de cualquier tipo sobre ciertos caracteres (desde pequeños pinchazos de alfiler hasta trazos a lápiz que marcan un mensaje oculto en un texto).
- Secuencia predefinida dentro de un texto.
- Afeitar la cabeza de un mensajero y tatuar en el cuero cabelludo el mensaje, dejando después que el crecimiento del pelo lo oculte.
- En imágenes digitales: sustituir el bit menos significativo de cada byte por los bits del mensaje que se desea ocultar.
- En archivos de audio, video, etc

Criptografía

Cifrado por sustitución
Algoritmo de César

- Se realiza siempre la misma sustitución: 1ª letra por 4ª; 2ª letra por 5ª; 3ª letra por 6ª... Es decir, la A en el mensaje original pasaría a ser la D en el mensaje cifrado.
- La expresión matemática de este algoritmo es:

$$C = (m + 3) \bmod L$$

donde C es el mensaje cifrado, m es el mensaje en claro, 3 sería la contraseña (que no es tal), L es el número de letras del alfabeto en cuestión. Esta expresión supone que cada letra esta asociada a un número (A=0, B=1, p. ej.).

Cifrado por sustitución
Algoritmo de César - Ejemplo

- Se desea cifrar el mensaje “SECRETO”
- Puede utilizarse una tabla para facilitar la conversión:

ABCDEFGHIJKLMNÑOPQRSTUVWXYZ
DEFGHIJKLMNÑOPQRSTUVWXYZABC

- El mensaje cifrado será “VHFUHWR”
- Es un algoritmo de cifrado **monoalfabético** porque a cada símbolo a cifrar le corresponde siempre el mismo símbolo cifrado.

Criptografía

Cifrado por transposición

Otra técnica de cifrado consiste, en vez de sustituir símbolos, en realizar permutaciones, es decir, cambiar su lugar. (ubicación, orden)

- **Rail Fence:**
  - El texto se escribe en diagonal hacia abajo en “rieles” de una valla imaginaria hasta el último riel, luego se escribe en diagonal hacia arriba y así sucesivamente. Luego se toma el mensaje por filas.

```
t v o u g d l r u i n
  e e l e o e a e n o
```

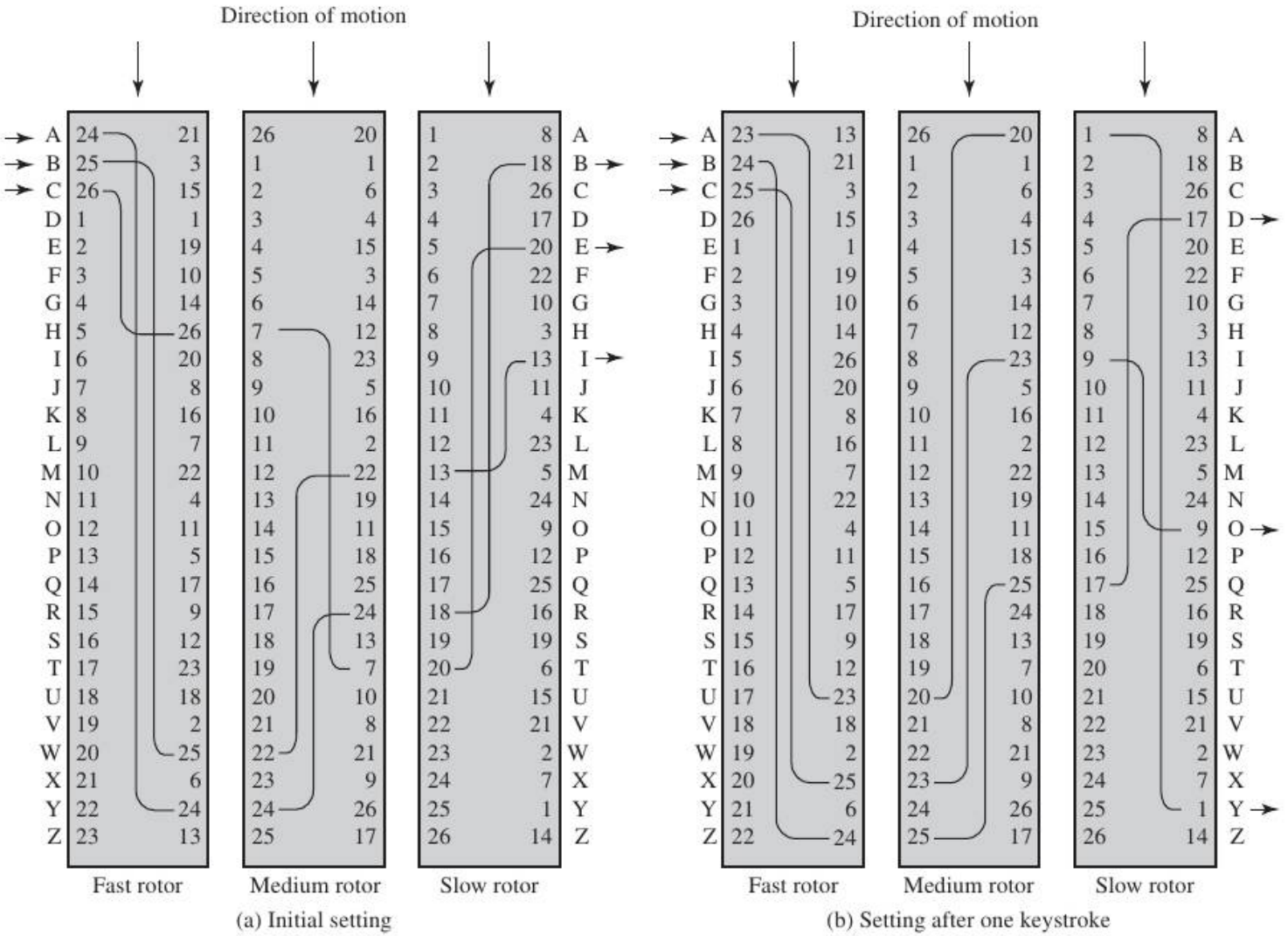
Mensaje → “te veo luego de la reunión”
← Rail Fence, Profundidad de 2
Texto cifrado → “tvougdlruineeleoeaeno”

- **Rotor Machine**
  - Compuesta por cilindros independientes con contactos eléctricos que implementan cada uno una sustitución monoalfabética.
  - Los cilindros giran a distinta velocidad, (como un odómetro*), logrando una sustitución polialfabética compleja.

* un “cuentakilómetros”

Criptografía

Rotor Machine



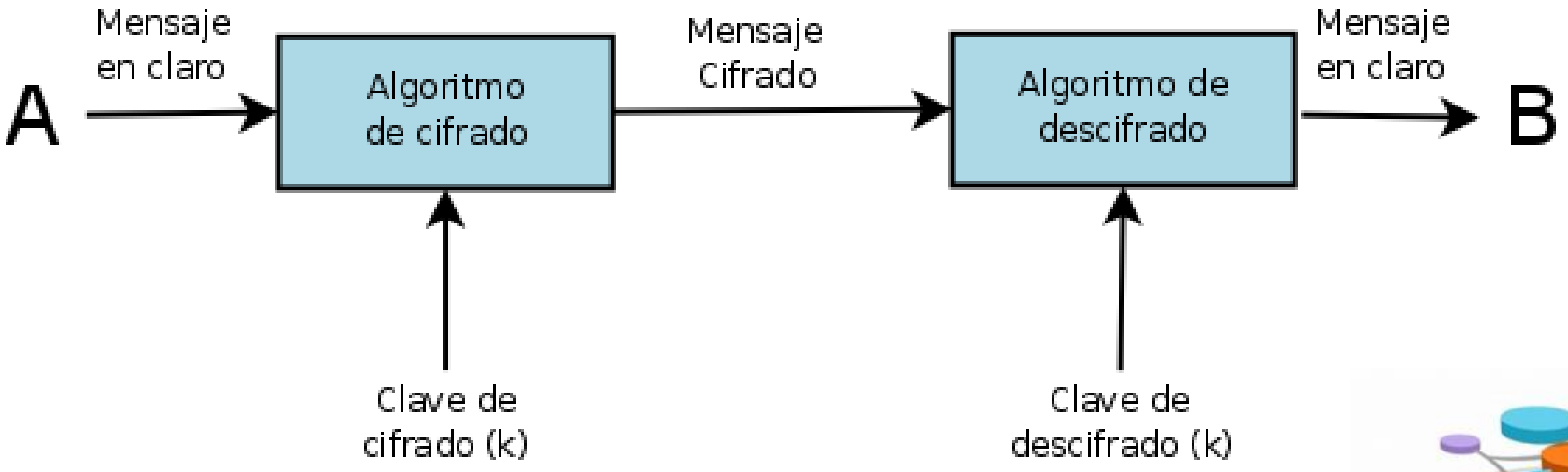
Enigma

https://summersidemakerspace.ca/projects/enigma-machine/

Criptografía Simétrica

Criptografía simétrica (convencional, o de clave secreta)

- El emisor cifra el mensaje con la clave *k* y se lo envía al receptor. Este último, que conoce dicha clave, la utiliza para descifrar la información.



Criptografía Simétrica

Criptografía simétrica (convencional, o de clave secreta)

- La criptografía simétrica se basa en la utilización de la misma clave para el cifrado y para el descifrado.
- La robustez de un algoritmo de cifrado simétrico recae en el conocimiento de dicha clave.
- Ventajas: sencillez de implementación, rapidez y robustez.
- **¿Qué desventajas se le ocurre que puede llegar a tener?**
- **¿Qué servicios de seguridad puede garantizar?**

Criptografía Simétrica

Criptografía simétrica (convencional, o de clave secreta)

- La criptografía simétrica se basa en la utilización de la misma clave para el cifrado y para el descifrado.
- La robustez de un algoritmo de cifrado simétrico recae en el conocimiento de dicha clave.
- Ventajas: sencillez de implementación, rapidez y robustez.
- Desventajas: Administración de claves no escalable.
- Puede garantizar:
  - Privacidad
  - Autenticidad
  - Integridad
- No Garantiza: No repudio

Criptografía Simétrica

Criptografía simétrica (convencional, o de clave secreta)

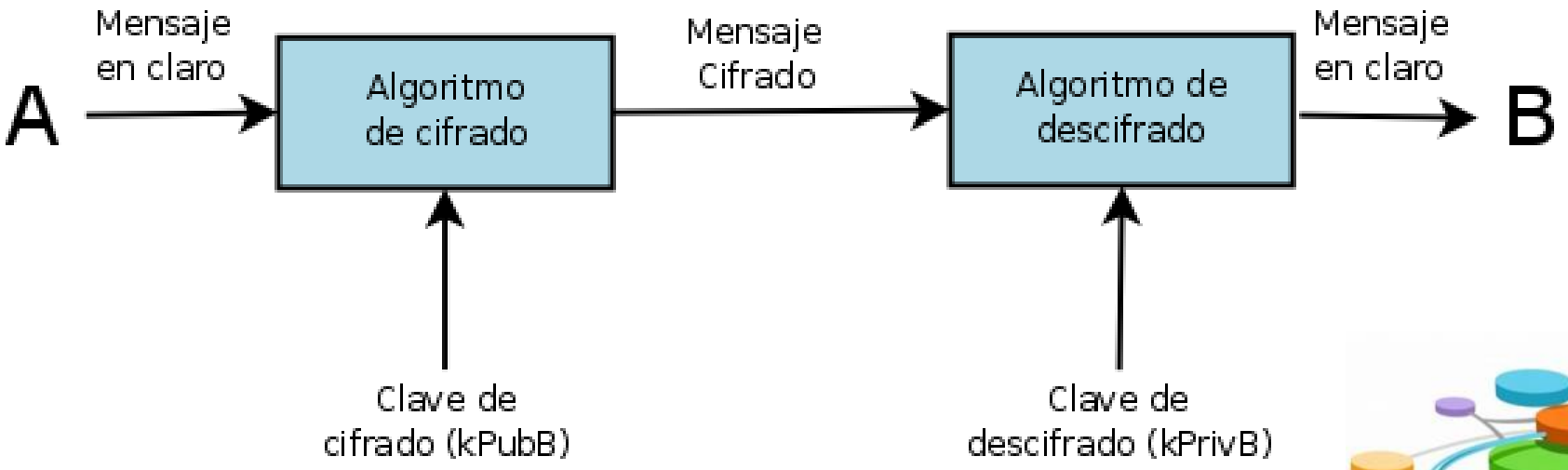
- Ejemplos de algoritmos de cifrado simétrico son:

Basados en Bloques	Basados en Stream
3DES (Triple-DES)	RC4
IDEA	Salsa20
Blowfish	ChaCha20
AES (Rijndael)	A5/1 - A5/2
CAST5	
Twofish	
Serpent	
Camellia	

Criptografía Asimétrica

Criptografía asimétrica (clave pública)

El usuario A cifra un mensaje con la clave pública del usuario B (destinatario), éste para descifrarlo utiliza su clave secreta correspondiente, únicamente conocida por él.



Criptografía Asimétrica

Criptografía asimétrica (clave pública)

- Se basa en la utilización de dos claves relacionadas, una para cifrar y otra para descifrar. (Denominadas clave pública y clave privada)
- La seguridad está basada en la dificultad de deducir una clave a partir del conocimiento de la otra. (la clave privada a partir de la clave pública)
- **¿Ventajas?**
- Desventajas: Mayor tiempo de procesamiento. **¿Otra desventaja?**

Criptografía Asimétrica

Criptografía asimétrica (clave pública)

- Se basa en la utilización de dos claves relacionadas, una para cifrar y otra para descifrar. (Denominadas clave pública y clave privada)
- La seguridad está basada en la dificultad de deducir una clave a partir del conocimiento de la otra. (la clave privada a partir de la clave pública)
- Ventajas: Mayor escalabilidad en la distribución de claves.
- Desventajas: Mayor tiempo de procesamiento. Necesidad de autenticar las claves públicas.

Criptografía Asimétrica

Criptografía asimétrica

¿Qué servicios de seguridad puede garantizar?

¿Conocen alguna implementación?

Criptografía Asimétrica

Criptografía asimétrica

- Puede garantizar:
  - Privacidad
  - Autenticidad
  - Integridad
  - No repudio

- Ejemplos de algoritmos de cifrado asimétrico son:
  - RSA (Rivest, Shamir y Adleman) 1977
  - DSA (Digital Signature Algorithm – Estandar FIPS) 1991
  - ECDSA (Elliptic Curve Digital Signature Algorithm)
  - ElGamal
  - Diffie-Hellman (intercambio de claves)
  - EdDSA (Edwards-curve Digital Signature Algorithm)
  - Ed25519

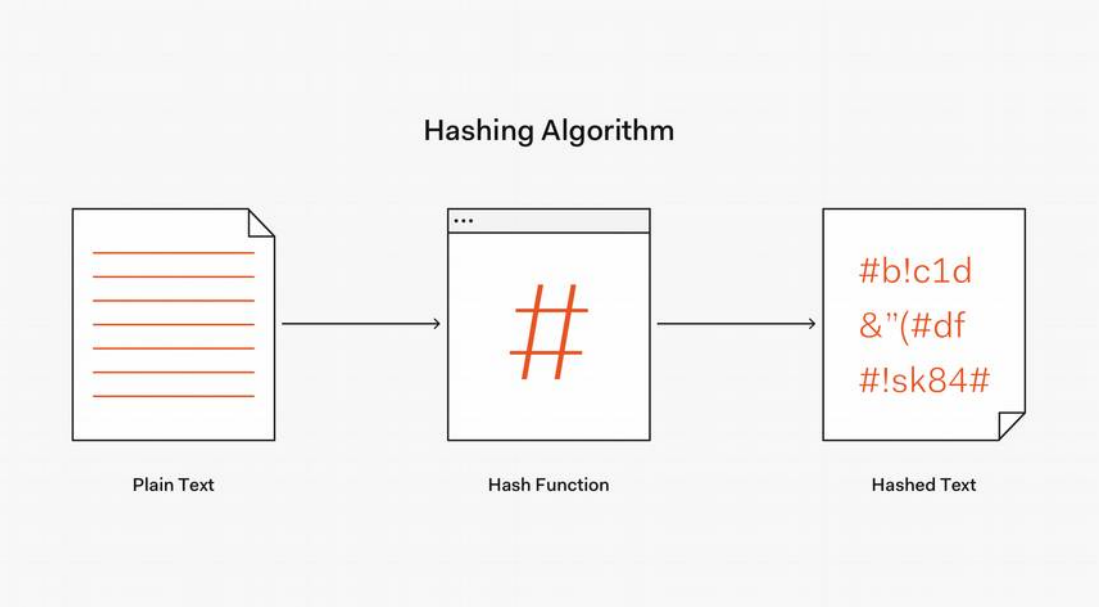
Integridad de Mensajes

Funciones HASH

¿Qué son?

Integridad de Mensajes

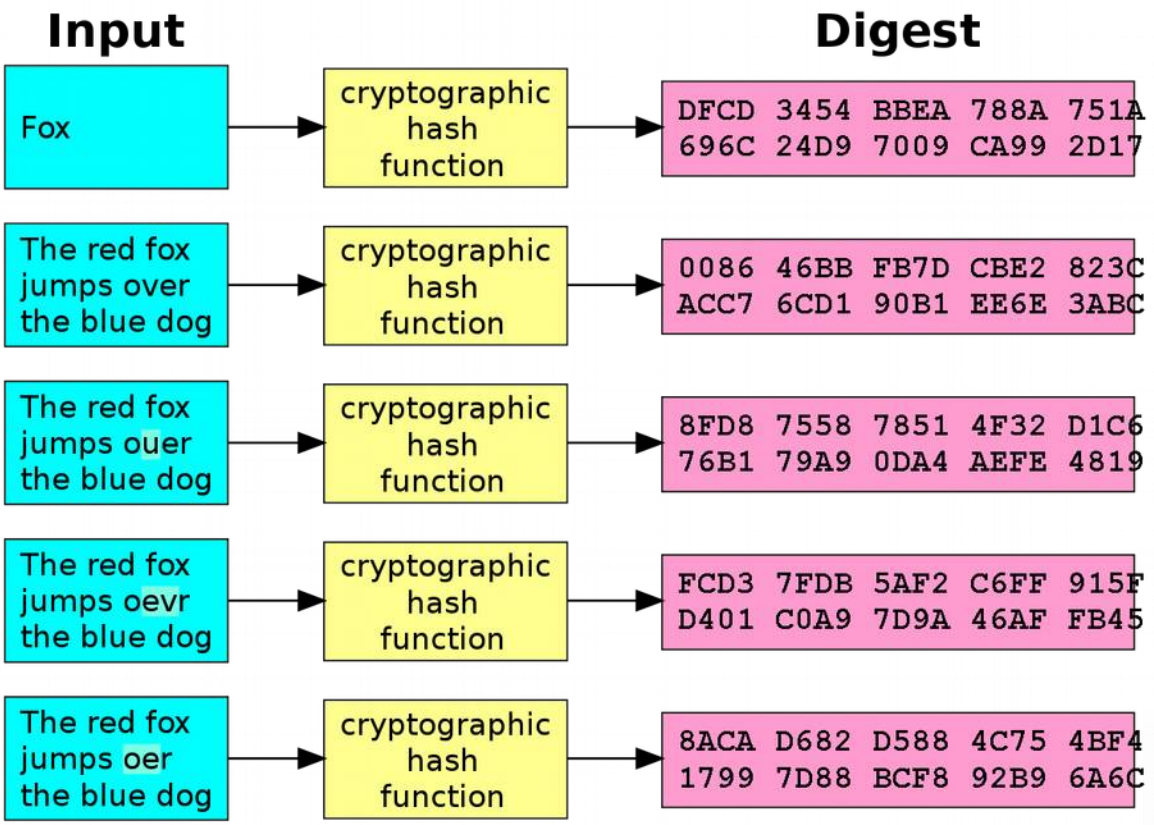
Funciones HASH



¿Para qué servicios de seguridad sirven?

Integridad de Mensajes

Funciones HASH Criptográficas



Integridad de Mensajes

Funciones HASH Criptográficas

- Una función hash criptográfica es una función computable que aplicada a un mensaje (m) de tamaño variable genera una representación de tamaño fijo del propio mensaje (H(m)).

- H(m) es mucho menor que m; por ejemplo, m puede tener una longitud de 1Mb, mientras que H(m) se puede reducir a 64 o 128 bits.

- Una función hash unidireccional es una función hash H de modo que para cualquier mensaje m es difícil encontrar un mensaje m' tal que H(m)=H(m'). Este tipo de función se denomina función resumen, y al valor H(m) se le suele llamar el resumen o digesto de m.

Integridad de Mensajes

Funciones HASH Criptográficas

- Algunas de las funciones hash más utilizadas son:

  - MD5 (Message Digest, MD) que genera firmas (digestos o resúmenes) de 128 bits.
  - SHA-1 (Secure Hash Algorithm), genera firmas de 160 bits.
  - SHA-2 (SHA-224, SHA-256, SHA-384, SHA-512).
  - RIPEMD-160, que genera firmas de 160 bits.
  - WHIRPOOL (firmas de 512 bits para mensajes menores a 2^{256} bits)
  - SHA-3 (Keccak)

Autenticación de Mensajes

Autenticación de mensajes

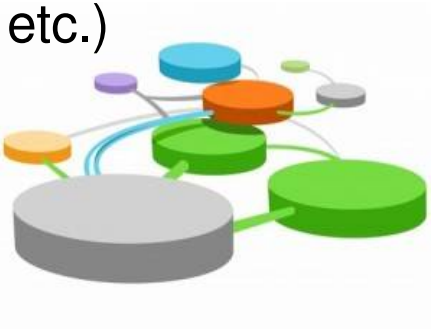
- Procedimiento para verificar que el mensaje recibido ha sido generado por la supuesta fuente que lo envía, y que no ha sido modificado.

- Adicionalmente, puede verificarse la secuencia y tiempo oportuno. (Que no ha ocurrido alteración en el orden de los mensajes, retraso o retransmisión).

Autenticación de Mensajes

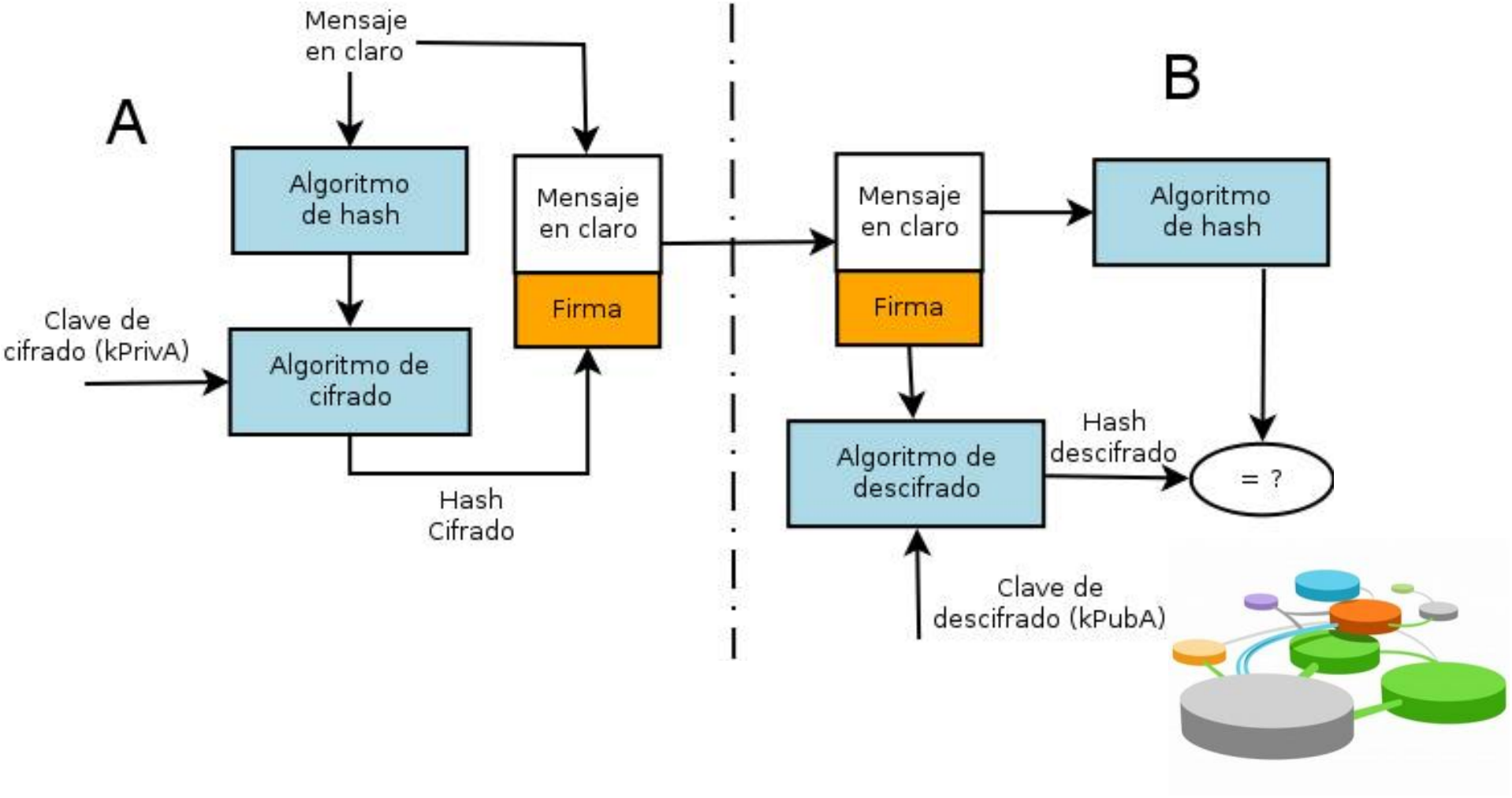
Código de Autenticación de Mensajes - MAC

- Una función MAC es una función que aplicada a un mensaje (m) de tamaño variable y una clave (k) genera una representación de tamaño fijo del propio mensaje $MAC=C_k(m)$.
- En resumen, es un mecanismo que provee autenticación e integridad a un mensaje.
- La clave secreta es compartida por emisor y receptor.
- Basado en cifrado simétrico por bloques. (Ej Data Authentication Algorithm)
- Basado en funciones hash (HMAC) o de cifrado (CMAC).
  - HMAC (RFC 2104):
    - MAC = H (k xor opad || H (k xor ipad || m))
- Donde H es una función Hash (MD5, SHA1, SHA256, SHA384, etc.)



Firma Digital

Firma digital



Firma digital

- Mecanismo de autenticación que permite al creador de un mensaje anexar un código que actúa como una firma, garantizando origen e integridad.

- Proceso de firmado:
  - El usuario A genera una huella digital H(m) del mensaje m y cifra dicha huella con su clave privada (kPrivA).
  - A continuación A envía al usuario B el mensaje sin cifrar (m) y su correspondiente resumen (H(m)) cifrado.
  - El usuario B obtiene la huella digital calculada por A utilizando la clave pública de A (kPubA) sobre el H(m) cifrado y a continuación genera la huella digital del mensaje enviado por el usuario A (H(m)').
  - B realiza una comparación de las dos huellas obtenidas. Si no coinciden (H(m) y H(m)') es que el mensaje o la huella enviada por A han sido modificados y por tanto la firma no es correcta.

Protección de comunicaciones

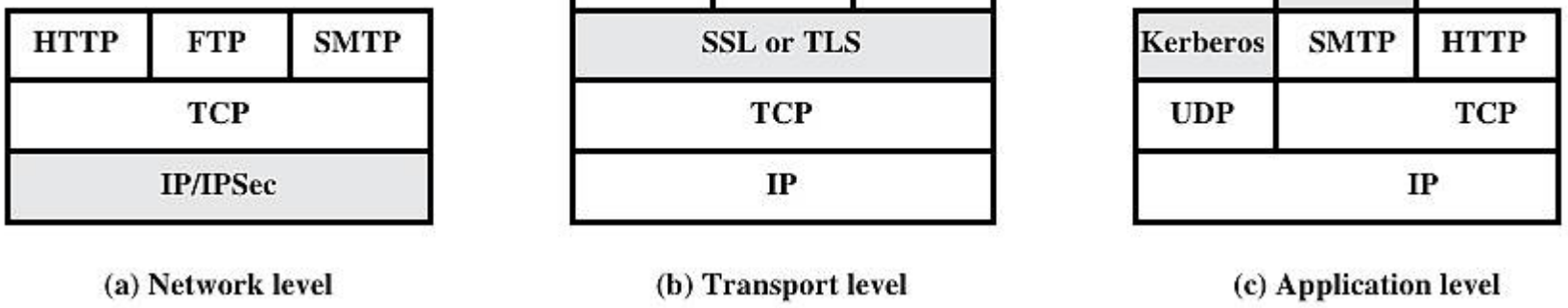
Fantástico, tenemos sistemas criptográficos. Ahora bien, ¿cómo utilizamos estas herramientas para dar **servicios de seguridad** a las comunicaciones?

¿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

- 2020 -

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)

- 2020 -

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)

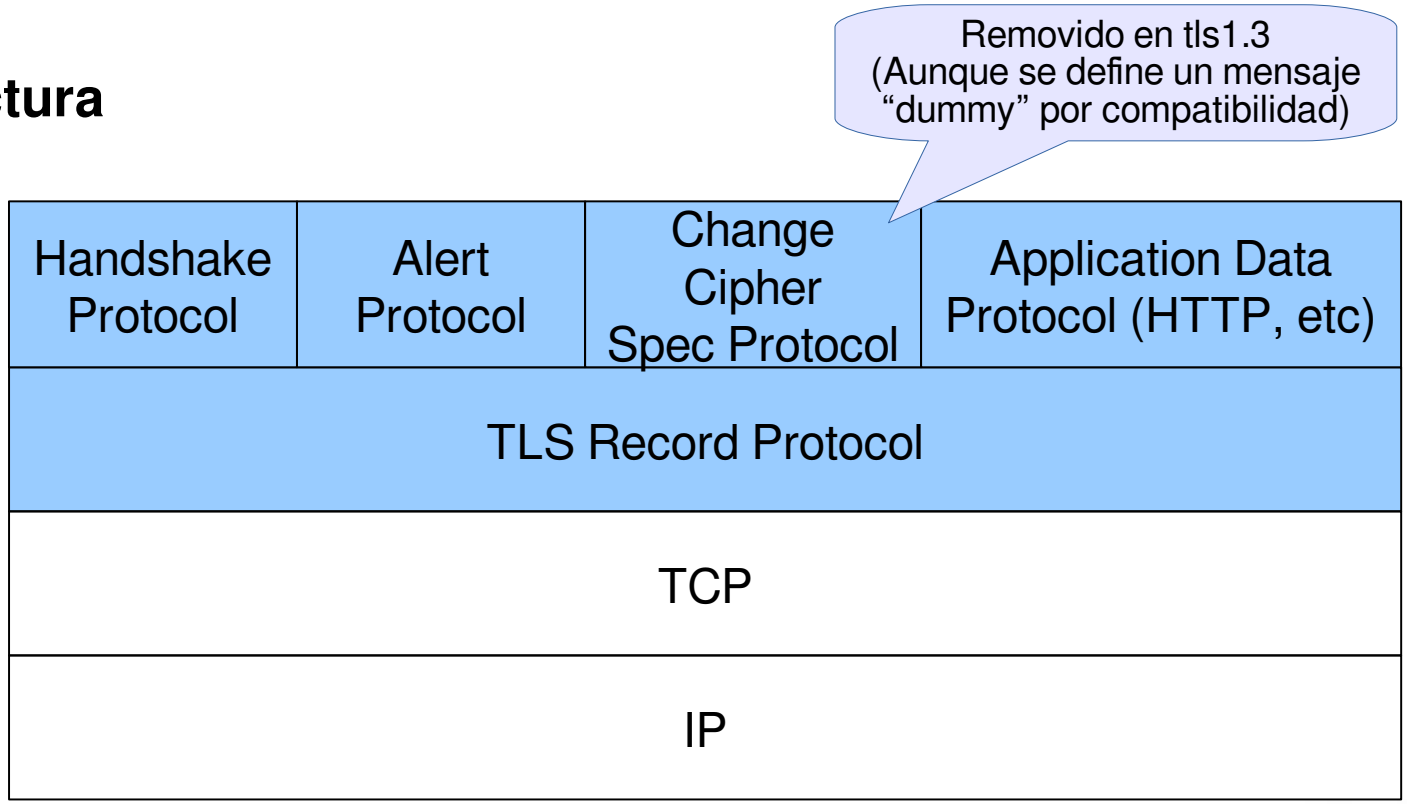
- Implementado sobre TCP proporciona seguridad a protocolos de aplicación como HTTP, SNMP, SIP, etc.

- 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

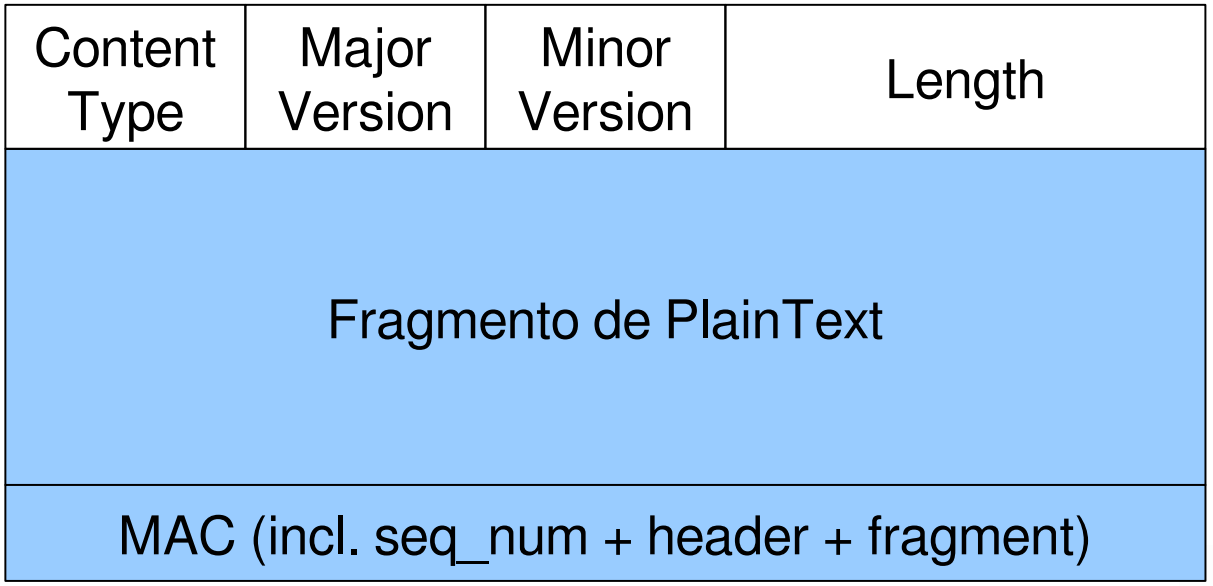
Transport Layer Security (TLS) Protocol
TLS Record Protocol

- **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
Alert Protocol

```
enum {
    close_notify(0),
    unexpected_message(10),
    bad_record_mac(20),
    decryption_failed_RESERVED(21),
    record_overflow(22),
    decompression_failure(30),
    handshake_failure(40),
    no_certificate_RESERVED(41),
    bad_certificate(42),
    unsupported_certificate(43),
    certificate_revoked(44),
    certificate_expired(45),
    certificate_unknown(46),
    illegal_parameter(47),
    unknown_ca(48),
    access_denied(49),
    decode_error(50),
    decrypt_error(51),
    export_restriction_RESERVED(60),
    protocol_version(70),
    insufficient_security(71),
    internal_error(80),
    user_canceled(90),
    no_renegotiation(100),
    unsupported_extension(110),
    (255)
} AlertDescription;
```

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)

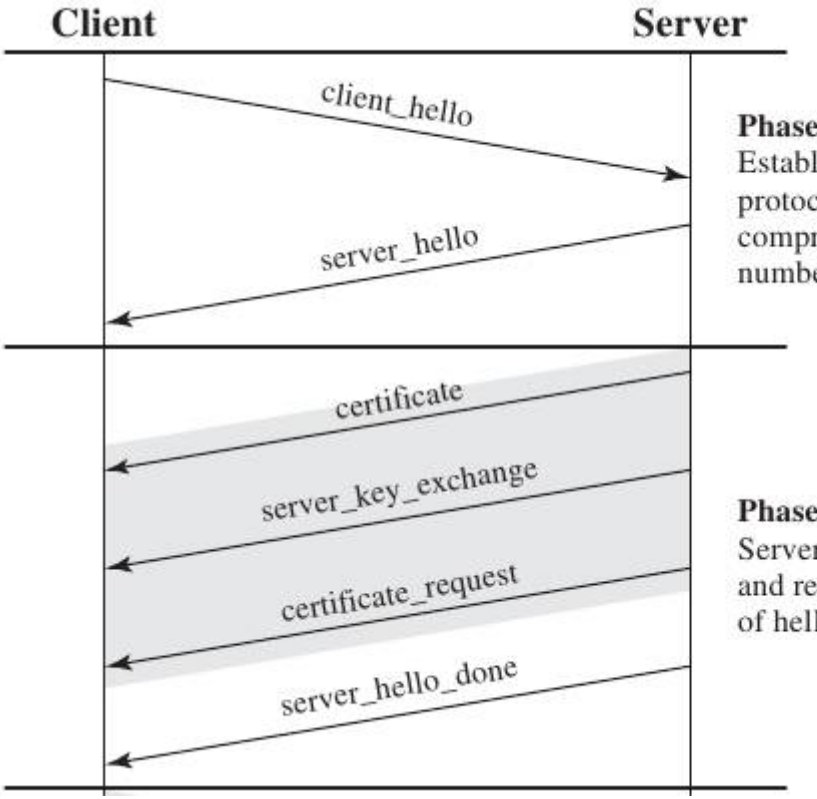
Transport Layer Security

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)

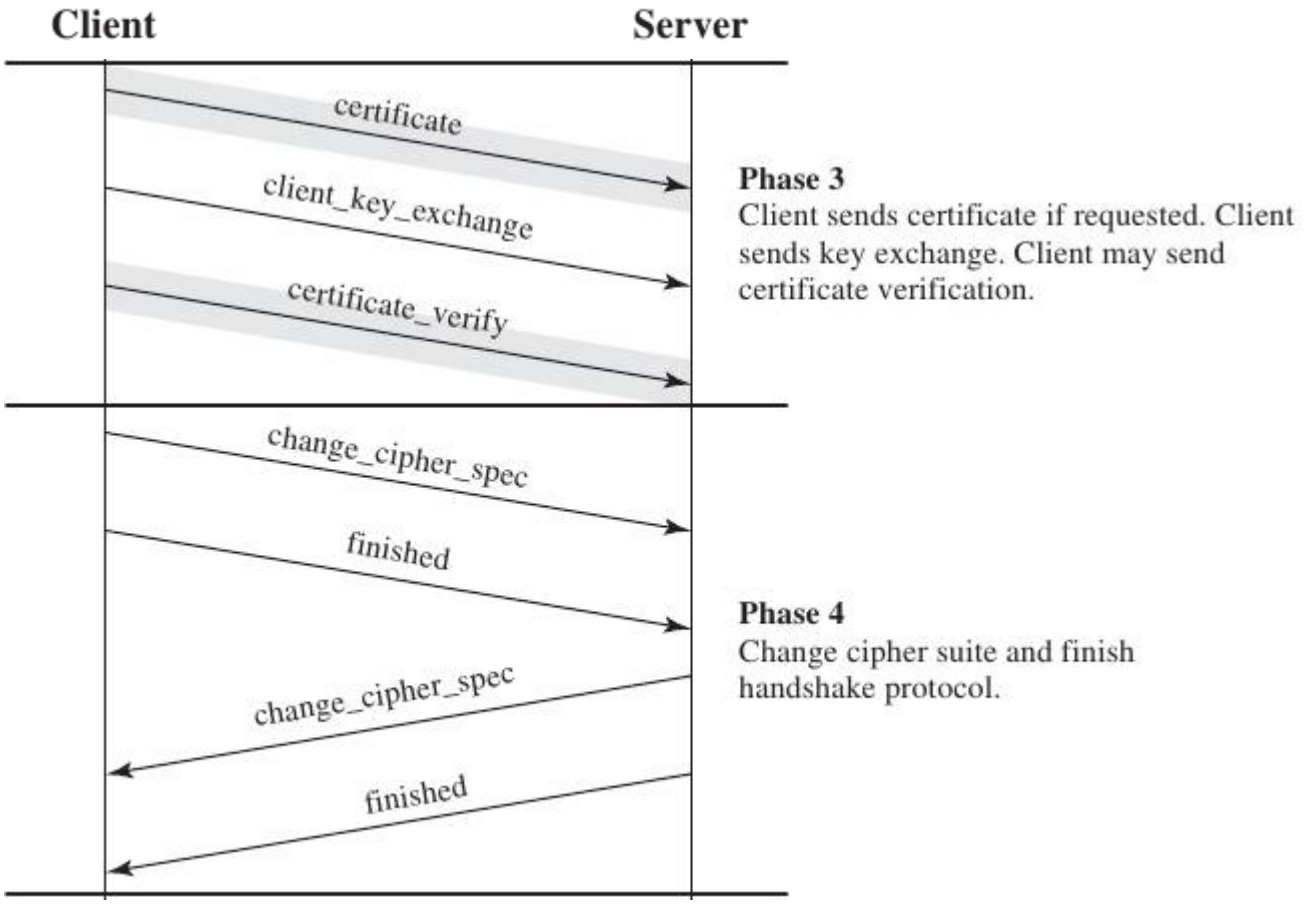


Phase 1
Establish security capabilities, including protocol version, session ID, cipher suite, compression method, and initial random numbers.

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

Transport Layer Security

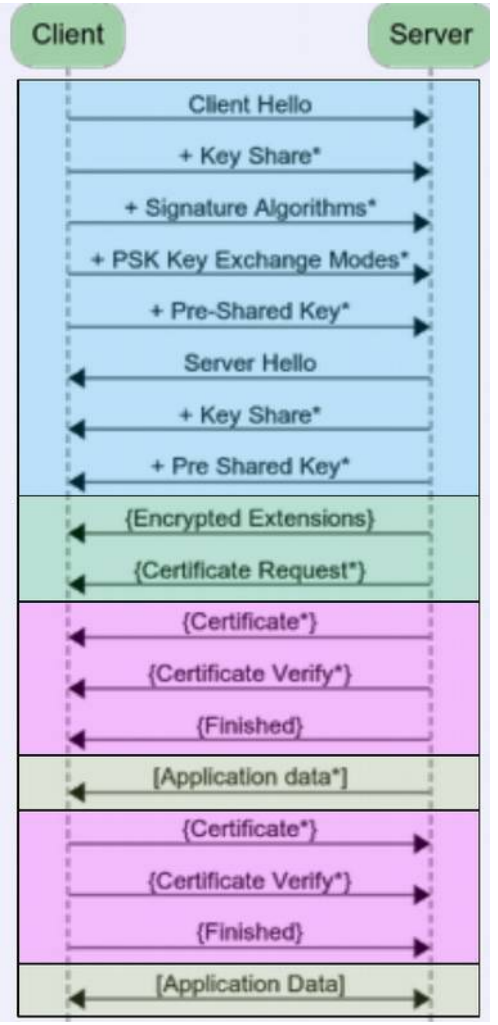
Transport Layer Security (TLS) Protocol
Handshake Protocol (v1.2)



Transport Layer Security

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

Key Exchange



Server Parameters

Authentication

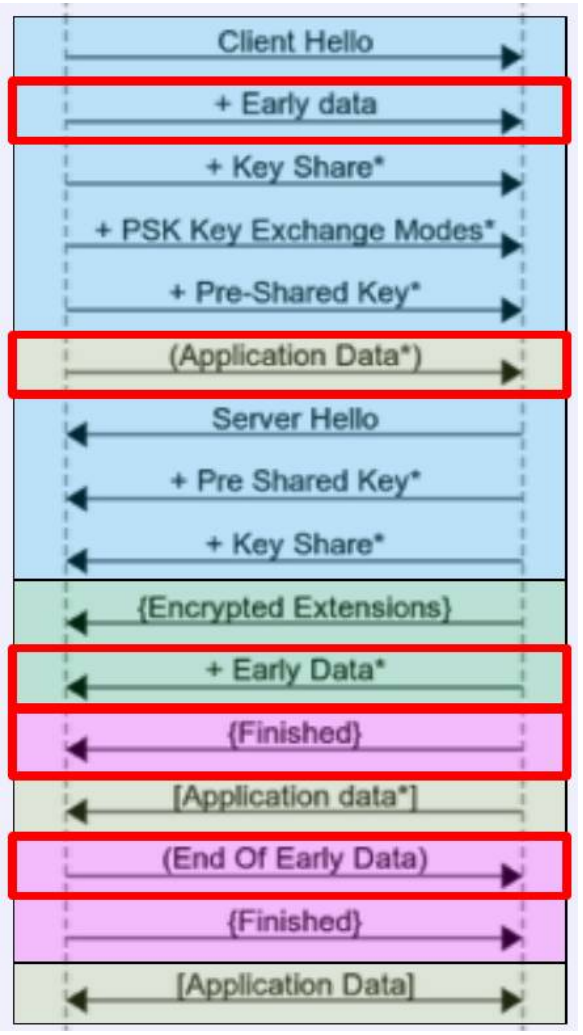
Transport Layer Security

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)

- 2020 -

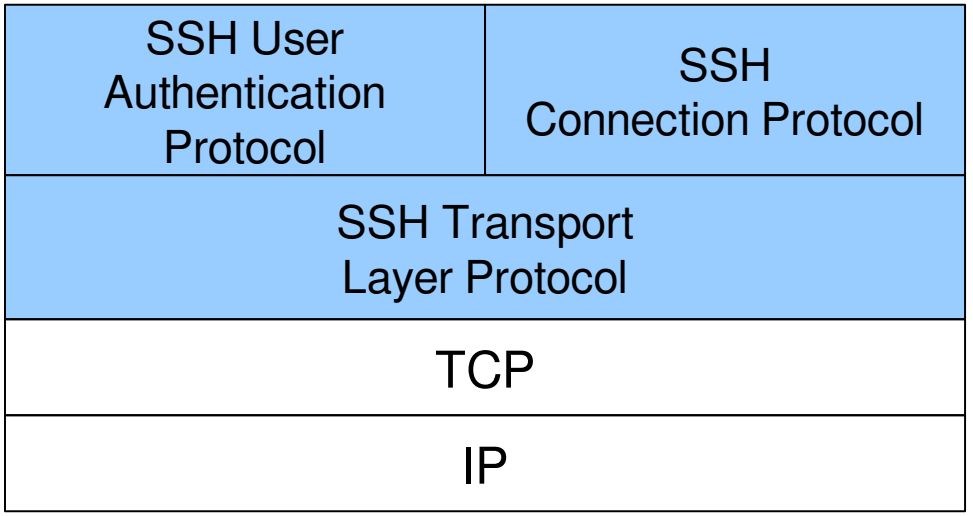
Secure Shell

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**:

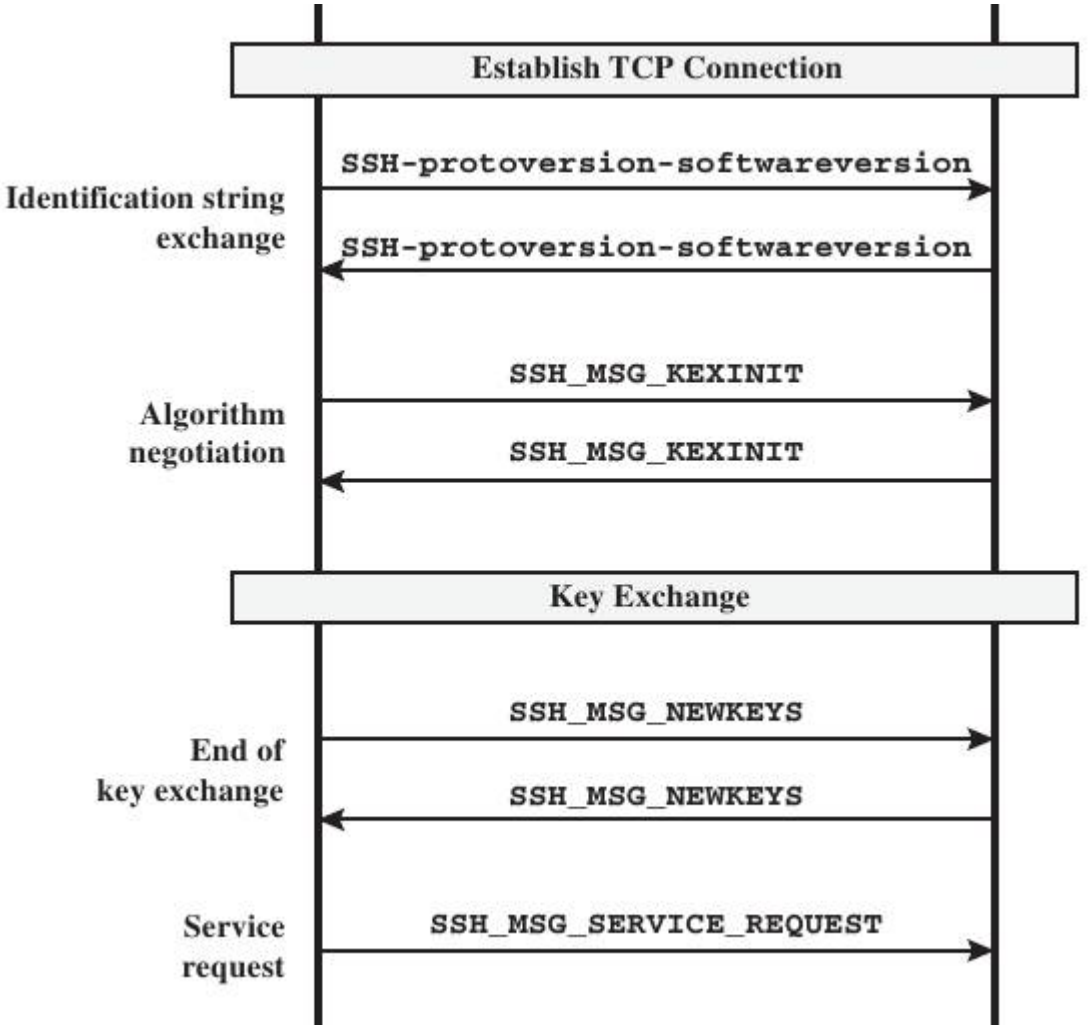
  Autentica usuario frente al servidor

- **SSH Connection Protocol**:

  Multiplexa múltiples canales de comunicación lógicos.

Secure Shell

SSH: Transport Layer Protocol



Secure Shell

SSH: User autenticacion protocol

Métodos de autenticación

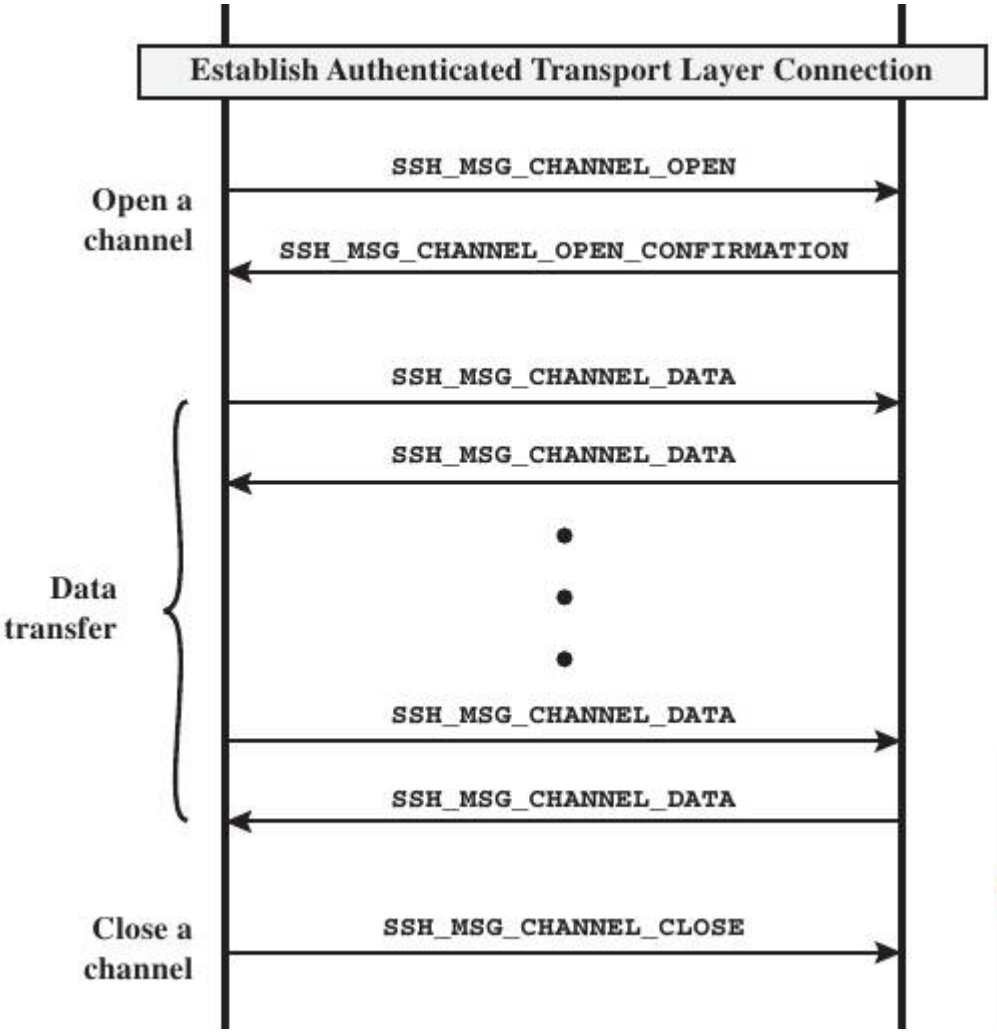
- RFC 4252
  - Clave pública (publickey)
  - Contraseña (password)
  - Basada en host (hostbased)
- RFC 4256
  - Intercambio de mensajes de autenticación genérico (interactivo)

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