یا ذالامن والامان

امنیت لایه انتقال:
SSL/TLS, SSH

مبتنی بر فصل ۶ از کتاب

Network Security Essentials

توسط: حمید رضا شهریاری

http://atlas.aut.ac.ir

فهرست مطالب

- خطرات تهدیدکننده لایه انتقال
- روشهای مختلف تامین امنیت لایه انتقال
  - SSL/TLS
  - SSH

خطرات تهدیدکننده لایه انتقال

- طراحی پروتکل TCP/IP بدون در نظر گرفتن هر گونه امنیت بوده

- نمونه ای از خطرات متداول:
  - ربایش نشست یا اتصال
  - شنود اطلاعات و داده‌ها
  - جعل آدرس (جعل کلاینت، سرور)

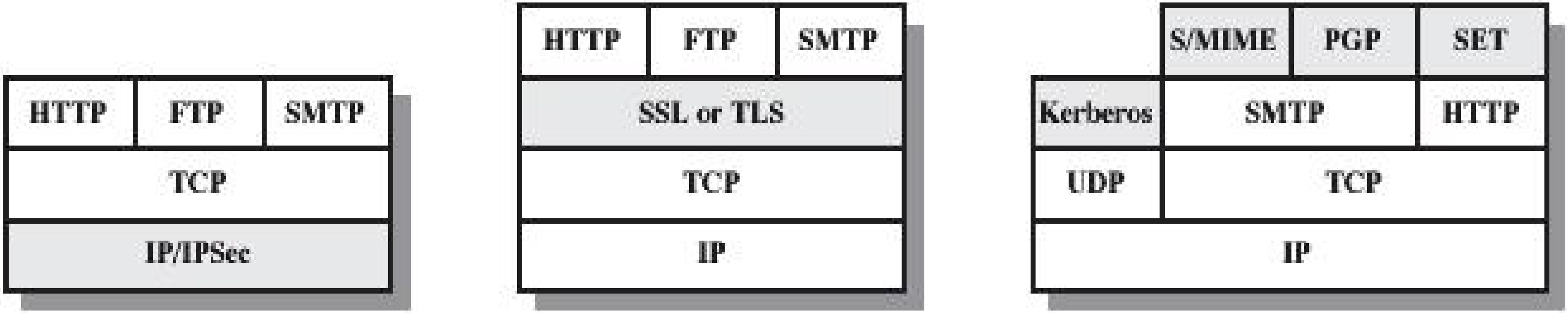
Threats on the Web

	Threats	Consequences	Countermeasures
Integrity	•Modification of user data •Trojan horse browser •Modification of memory •Modification of message traffic in transit	•Loss of information •Compromise of machine •Vulnerabilty to all other threats	Cryptographic checksums
Confidentiality	•Eavesdropping on the net •Theft of info from server •Theft of data from client •Info about network configuration •Info about which client talks to server	•Loss of information •Loss of privacy	Encryption, Web proxies
Denial of Service	•Killing of user threads •Flooding machine with bogus requests •Filling up disk or memory •Isolating machine by DNS attacks	•Disruptive •Annoying •Prevent user from getting work done	Difficult to prevent
Authentication	•Impersonation of legitimate users •Data forgery	•Misrepresentation of user •Belief that false information is valid	Cryptographic techniques

روشهای مختلف تامین امنیت وب

- استفاده از IPSec . مزایا :
  - همه منظوره
  - شفاف از دید کاربران لایه بالاتر
- استفاده از SSL/TLS
  - شفاف از دید برنامه‌های کاربردی سطح بالاتر
  - پشتیبانی مرورگرهای متداول و نیز وب سرورها
- سرویسهای امنیتی وابسته به کاربرد خاص
  - مانند PGP

Web Security Approaches



HTTP FTP SMTP
TCP
IP/IPSec

(a) Network Level

HTTP FTP SMTP
SSL or TLS
TCP
IP

(b) Transport Level

S/MIME PGP SET
Kerberos SMTP HTTP
UDP TCP
IP

(c) Application Level

SSL – تاریخچه

- July, 1994:
  - شرکت Netscape طراحی SSL 1.0 را انجام داد.
- Dec, 1994
  - مرورگر Netscape همراه با SSL 2.0 به بازار عرضه شد.
  - آسیب پذیر بود. کمتر از ۱ ساعت می شد به آن نفوذ کرد.
  - محدودیت استفاده از کلیدهای ٤٠ بیتی در خارج آمریکا
- Nov, 1995
  - شرکت Netscape توصیف SSL 3.0 را منتشر کرد
  - با تغییرات و جهش عمده نسبت به نسخه های قبلی همراه بود

SSL/TLS – تاریخچه(...ادامه)

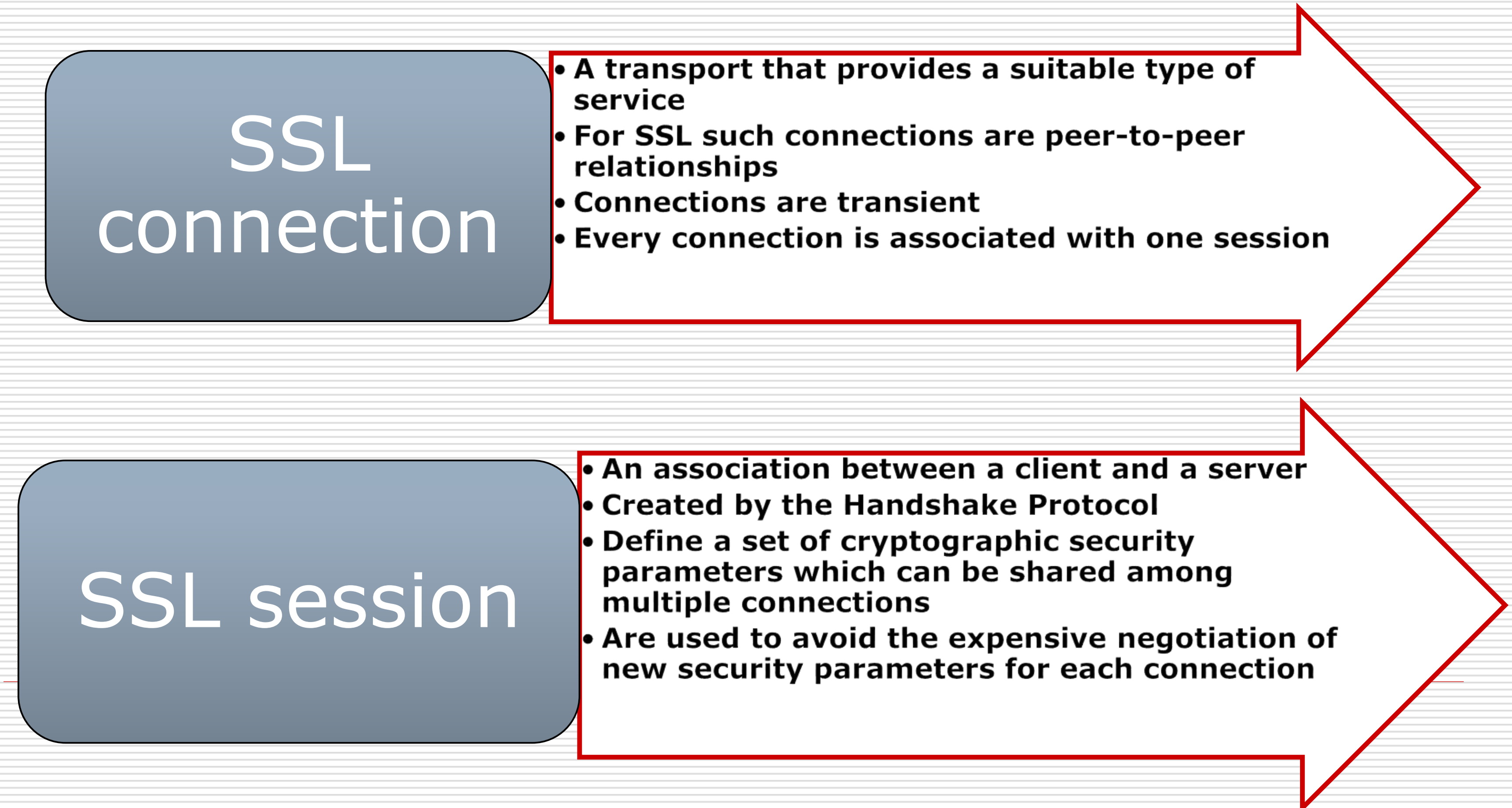
- Jan, 1999
  - TLS 1.0 بطور رسمی همراه با RFC 2246 به بازار عرضه شد.
  - در واقع همان SSL v3.1 بود که به دلایل تجاری تغییر نام داده بود.
- April, 2006: معرفی TLS 1.1
- August, 2008: معرفی TLS 1.2
- August 2018: معرفی TLS 1.3

SSL/TLS

- □ لایه امنیتی در بالای لایه انتقال
- □ ارائه شده توسط شرکت Netscape
- □ سرویس قابل اطمینان انتها به انتها (end to end) و مبتنی بر TCP
- □ پروتکل آن در دو لایه پیاده سازی می شود

SSL/TLS – معماری

- □ دو مفهوم اساسی در SSL:



SSL connection

- **A transport that provides a suitable type of service**
- **For SSL such connections are peer-to-peer relationships**
- **Connections are transient**
- **Every connection is associated with one session**

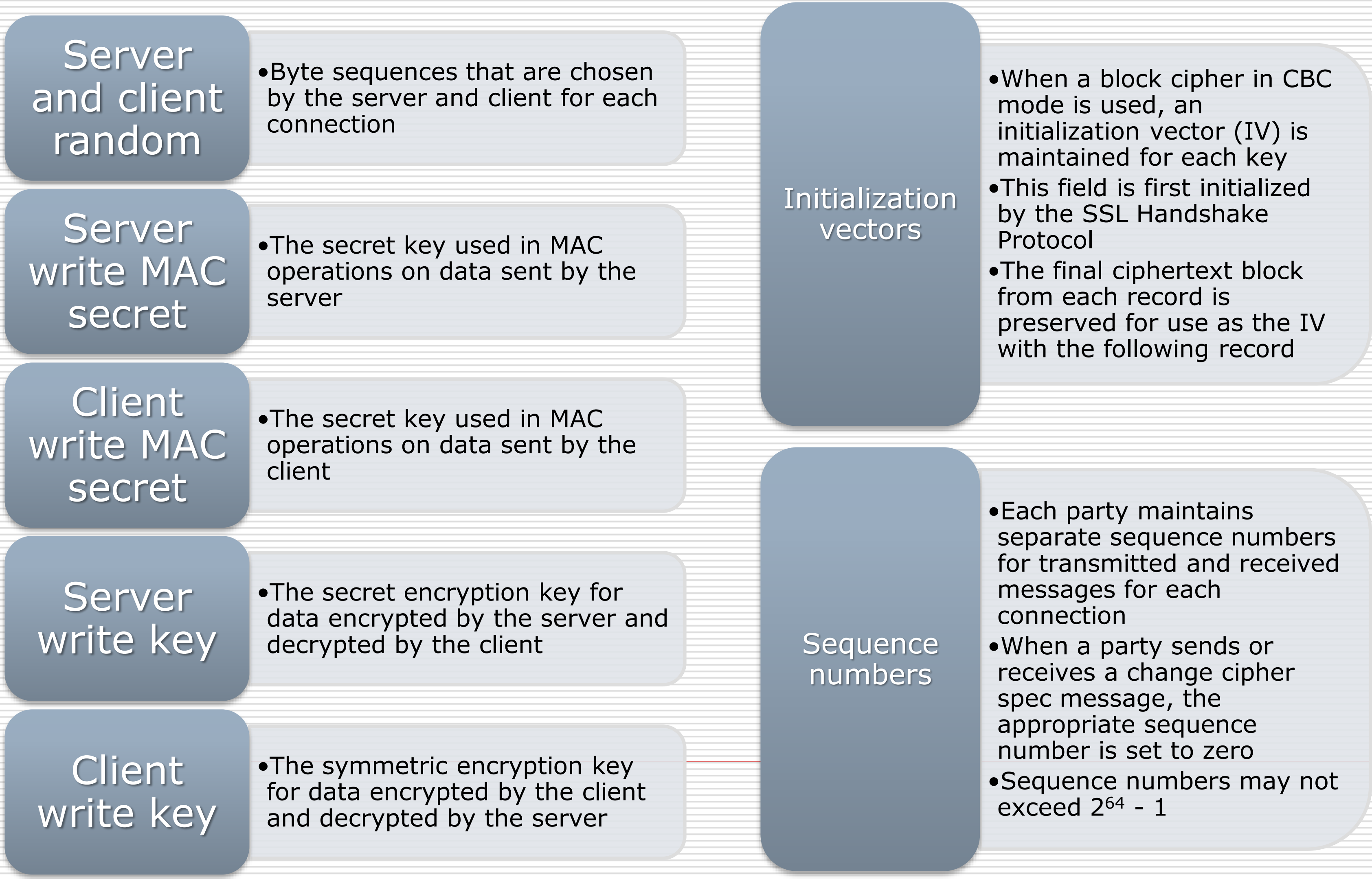
SSL session

- **An association between a client and a server**
- **Created by the Handshake Protocol**
- **Define a set of cryptographic security parameters which can be shared among multiple connections**
- **Are used to avoid the expensive negotiation of new security parameters for each connection**

یک نشست SSL/TLS با پارامترهای زیر مشخص می‌شود

Session identifier	An arbitrary byte sequence chosen by the server to identify an active or resumable session state
Peer certificate	An X509.v3 certificate of the peer; this element of the state may be null
Compression method	The algorithm used to compress data prior to encryption
Cipher spec	Specifies the bulk data encryption algorithm and a hash algorithm used for MAC calculation; also defines cryptographic attributes such as the hash_size
Master secret	48-byte secret shared between the client and the server
Is resumable	A flag indicating whether the session can be used to initiate new connections

یک حالت اتصال با پارامترهای زیر مشخص می‌شود



Server and client random
- Byte sequences that are chosen by the server and client for each connection

Server write MAC secret
- The secret key used in MAC operations on data sent by the server

Client write MAC secret
- The secret key used in MAC operations on data sent by the client

Server write key
- The secret encryption key for data encrypted by the server and decrypted by the client

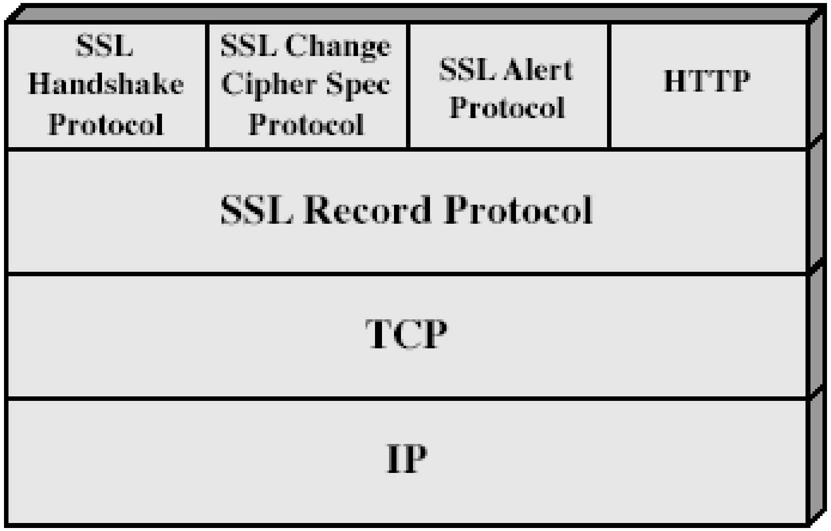
Client write key
- The symmetric encryption key for data encrypted by the client and decrypted by the server

Initialization vectors
- When a block cipher in CBC mode is used, an initialization vector (IV) is maintained for each key
- This field is first initialized by the SSL Handshake Protocol
- The final ciphertext block from each record is preserved for use as the IV with the following record

Sequence numbers
- Each party maintains separate sequence numbers for transmitted and received messages for each connection
- When a party sends or receives a change cipher spec message, the appropriate sequence number is set to zero
- Sequence numbers may not exceed $2^{64} - 1$

SSL – معماری

- □ لایه اول بالای لایه انتقال و لایه دوم در لایه کاربرد
- □ لایه اول شامل پروتکل Record و لایه دوم مربوط به سرویسهای مدیریتی بوده و شامل پروتکلهای زیر می شود



SSL Handshake Protocol

SSL Change Cipher Spec Protocol

SSL Alert Protocol

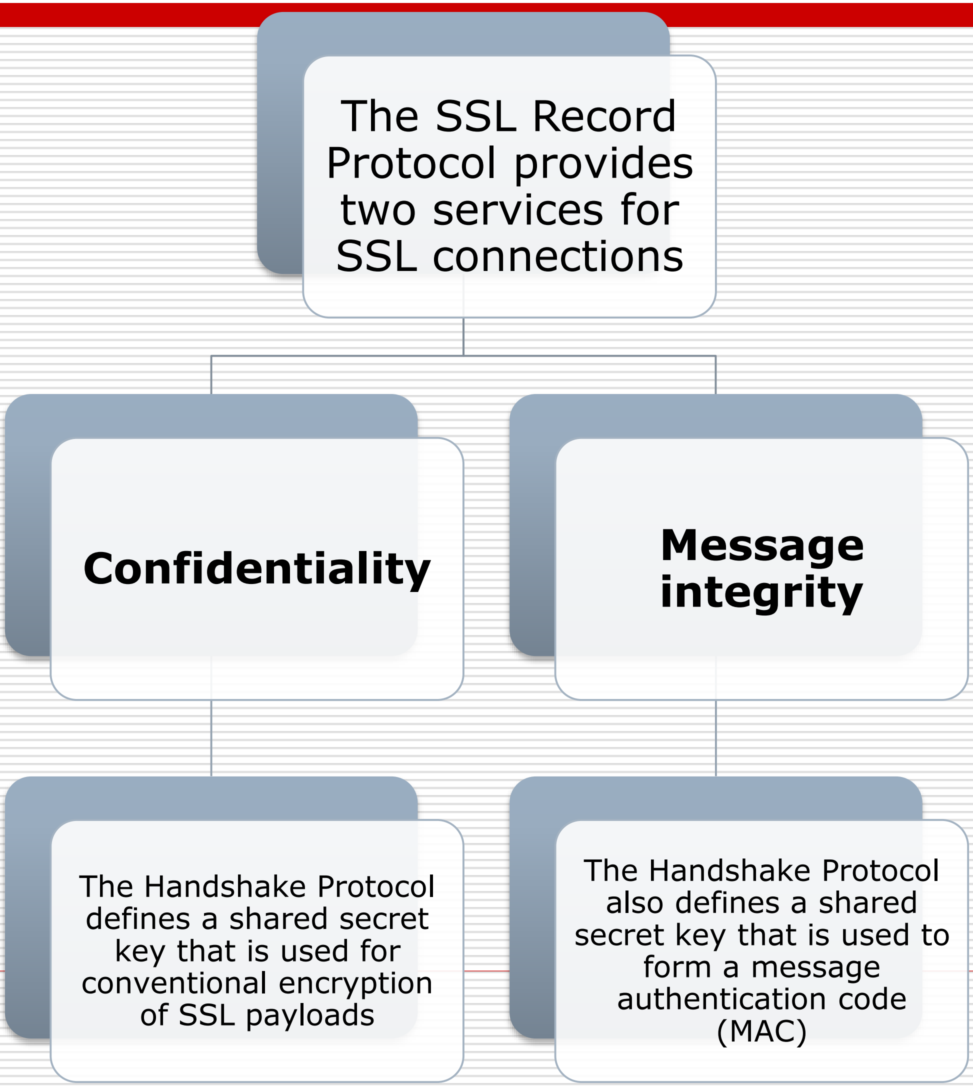
HTTP

SSL Record Protocol

TCP

IP

SSL Record Protocol



The SSL Record Protocol provides two services for SSL connections

Confidentiality

The Handshake Protocol defines a shared secret key that is used for conventional encryption of SSL payloads

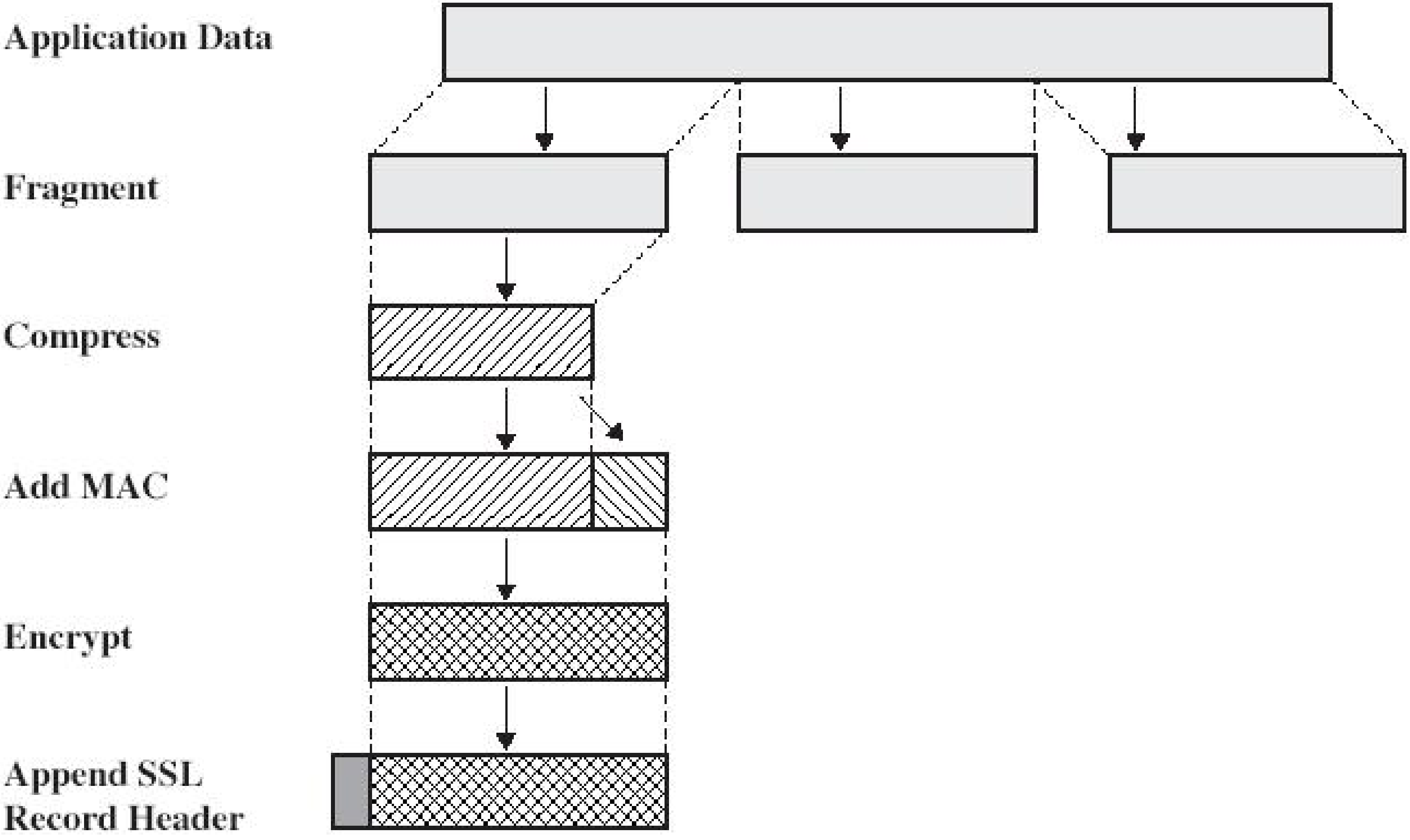
Message integrity

The Handshake Protocol also defines a shared secret key that is used to form a message authentication code (MAC)

SSL – پروتکلها

- SSL Record Protocol : دو سرویس برای SSL فراهم می کند:
  - محرمانگی :
    - با استفاده از یک کلید متقارن مخفی که در پروتکل Handshake توافق شده است.
    - استفاده از یکی از الگوریتم‌های رمزنگاری متقارن توافق شده
    - صحت پیغام
    - تولید MAC با استفاده از کلید متقارن مخفی
    - استفاده از تابع Hash توافق شده
- وظیفه تولید و توزیع کلیدهای متقارن برای انجام رمزگذاری مرسوم و نیز محاسبه MAC برعهده پروتکل handshake است

Record Protocol Operation



SSL – پروتکل‌ها

اعمال انجام شده در پروتکل Record

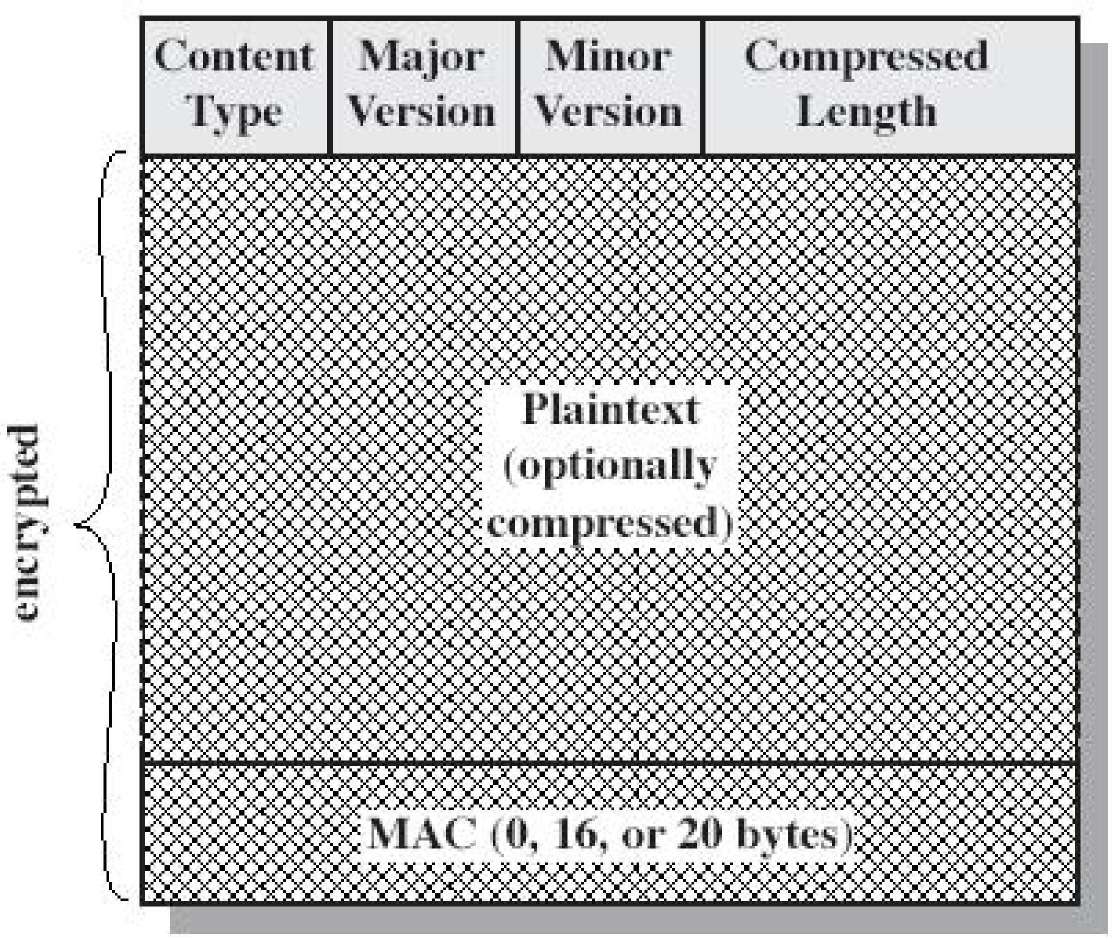
- قطعه بندی: تولید بلاکهای به طول 2^{14} یا کمتر .
- فشرده سازی : اختیاری و بدون از دست رفتن داده.
- تولید MAC : مشابه HMAC و روی ورودی زیر انجام می گیرد:
  - □ (محتوای بلاک، طول بلاک، نوع فشرده سازی، شماره سریال)

- رمزنگاری : استفاده از رمز بلاکی یا نهری.
- اضافه کردن سرآیند : به ابتدای بلاک رمزشده می چسبد و شامل موارد زیر است:

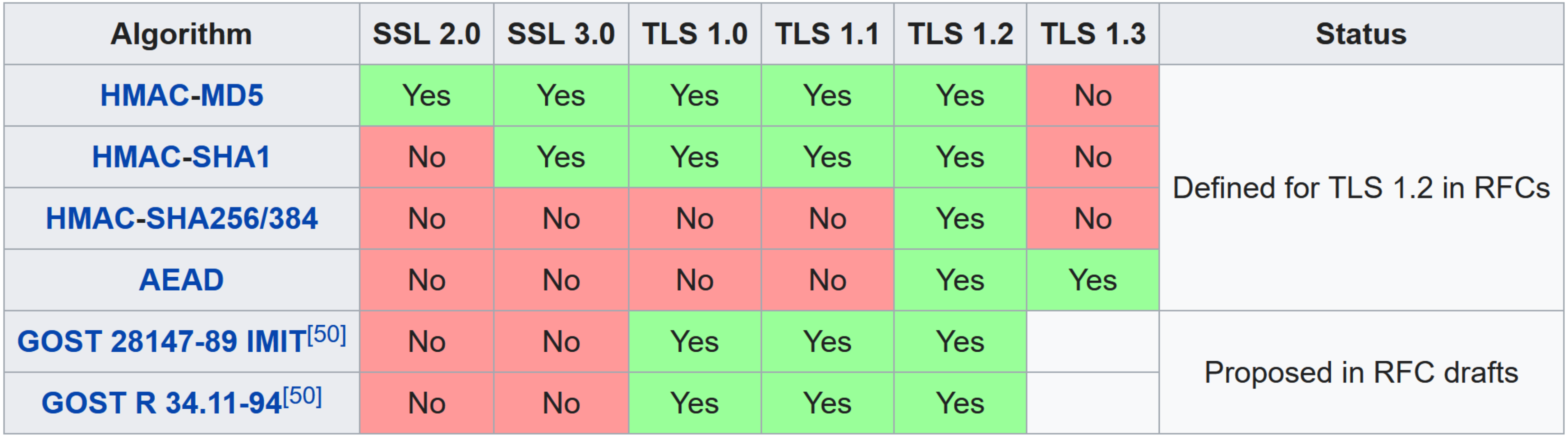
(نوع محتوا، نسخه اصلی SSL، نسخه فرعی SSL، طول داده فشرده شده)

نوع محتوا(Content Type) بیان کننده پروتکل استفاده کننده از این سرویس در لایه دوم می باشد.

SSL Record Format



Data Integrity in SSL/TLS



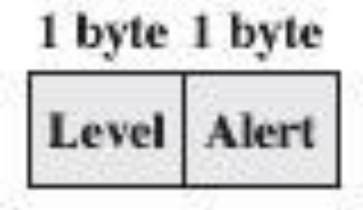
Algorithm	SSL 2.0	SSL 3.0	TLS 1.0	TLS 1.1	TLS 1.2	TLS 1.3	Status
HMAC-MD5	Yes	Yes	Yes	Yes	Yes	No	Defined for TLS 1.2 in RFCs
HMAC-SHA1	No	Yes	Yes	Yes	Yes	No	
HMAC-SHA256/384	No	No	No	No	Yes	No	
AEAD	No	No	No	No	Yes	Yes	
GOST 28147-89 IMIT[50]	No	No	Yes	Yes	Yes		Proposed in RFC drafts
GOST R 34.11-94[50]	No	No	Yes	Yes	Yes		

SSL – پروتکلها

پروتکل SSL Alert:

- هشدارها و خطاهای مربوط به SSL را به طرف مقابل منتقل می کند
- شدت خطای پیش آمده : Warning or Fatal
- مانند بقیه داده های SSL فشرده سازی و رمزنگاری می شود.
- نمونه خطاها :

unexpected message, bad record mac, decompression failure, handshake failure



1 byte	1 byte
Level	Alert

SSL – پروتکلها

پروتکل SSL Handshake

- پیش از انتقال هر نوع داده ای تحت SSL انجام می شود.
- با استفاده از آن کارفرما و کارگزار می توانند :
  - □ همدیگر را شناسایی کنند
  - □ الگوریتم های رمزنگاری، توابع درهم ساز مورد استفاده و کلیدهای رمزنگاری متقارن و نامتقارن را رد و بدل کنند.

قرارداد توافق – فاز Hello

پروتکل SSL Handshake

شامل ٤ فاز اصلی زیر می باشد

- □ مشخص کردن قابلیتهای رمزنگاری دو طرف
- □ احراز هویت کارگزار به کارفرما و مبادله کلیدهای آن
- □ احراز هویت کارفرما به کارگزار و مبادله کلیدهای آن
- □ جایگزینی پارامترهای رمزنگاری جدید به جای قبلی و خاتمه توافق

قرارداد توافق – فاز Hello

ارسال پیغام Hello توسط کارفرما (آغازگر جلسه)

- پیشنهاد نسخه قرارداد: آخرین نسخه پشیبانی شده توسط کارفرما
- پیشنهاد الگوریتم‌های مناسب و روش تبادل کلید آنها
- پیشنهاد روش فشرده سازی مناسب
- انتخاب نسخه و الگوریتم های مورد قبول کارگزار
- کارگزار بررسی می‌کند که آیا این پیشنهاد قابل قبول است یا نه؟

قرارداد توافق – فاز تبادل کلید

- ارسال گواهی کارگزار برای کارفرما
  - همراه با کلید عمومی(RSA) یا پارامترهای DH

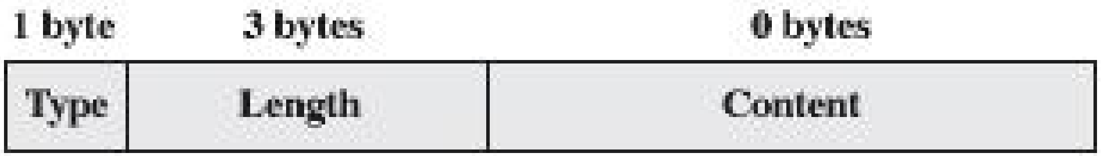
- تولید و ارسال کلید مخفی (سرّی)
  - کارفرما کلید سرّی را تولید کرده و برای کارگزار می‌فرستد
  - یا این که هر دو با استفاده از پارامترهای DH کلید سرّی را محاسبه می‌کنند.

قرارداد توافق – فاز خاتمه

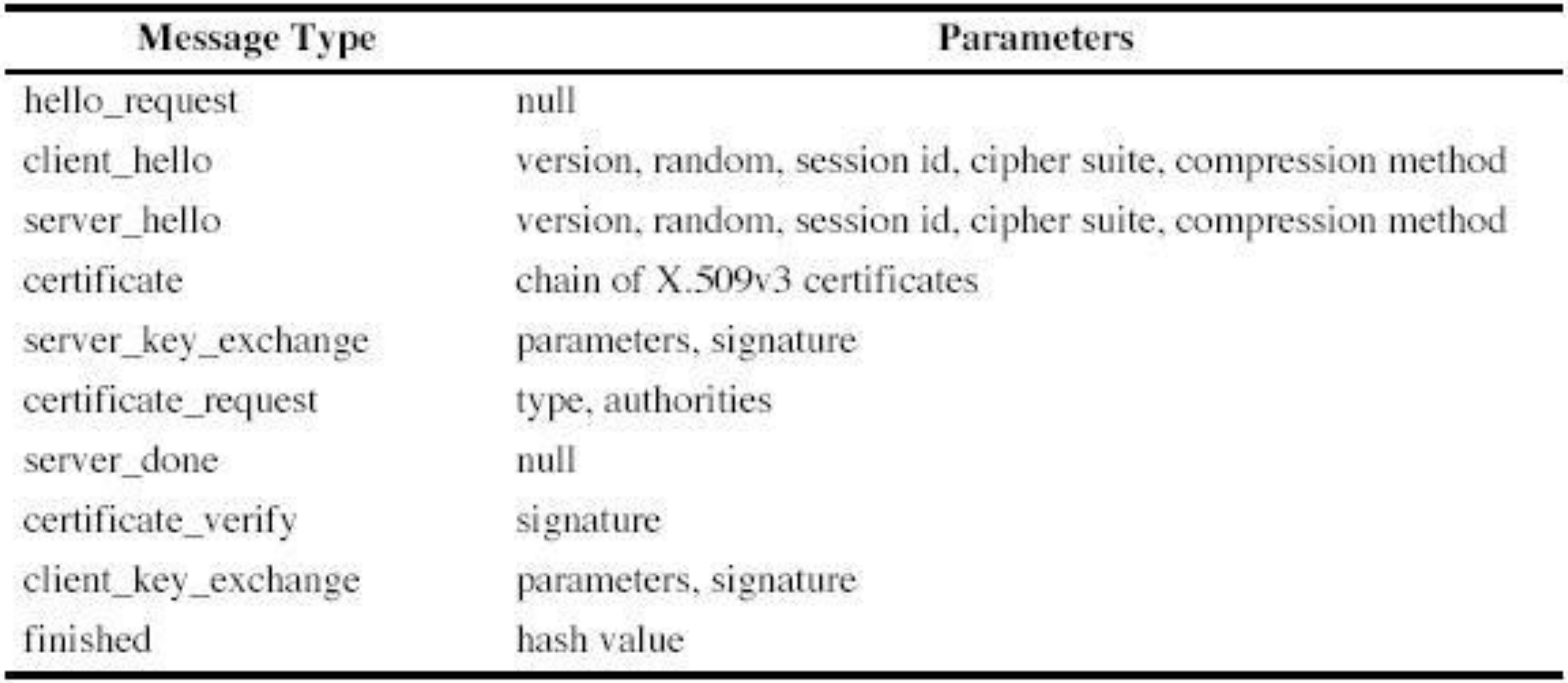
- فعال کردن قرارداد تغییر مشخصات رمز
  - کارفرما قرارداد تغییر مشخصات رمز را فعال کرده و برای کارگزار می فرستد.
  - کارگزار نیز قرارداد تغییر مشخصات رمز را فعال کرده و ارسال می کند.

- پایان
  - ارسال پیغام پایانی
  - آغاز تبادل اطلاعات به صورت محرمانه و با پارامترهای جدید

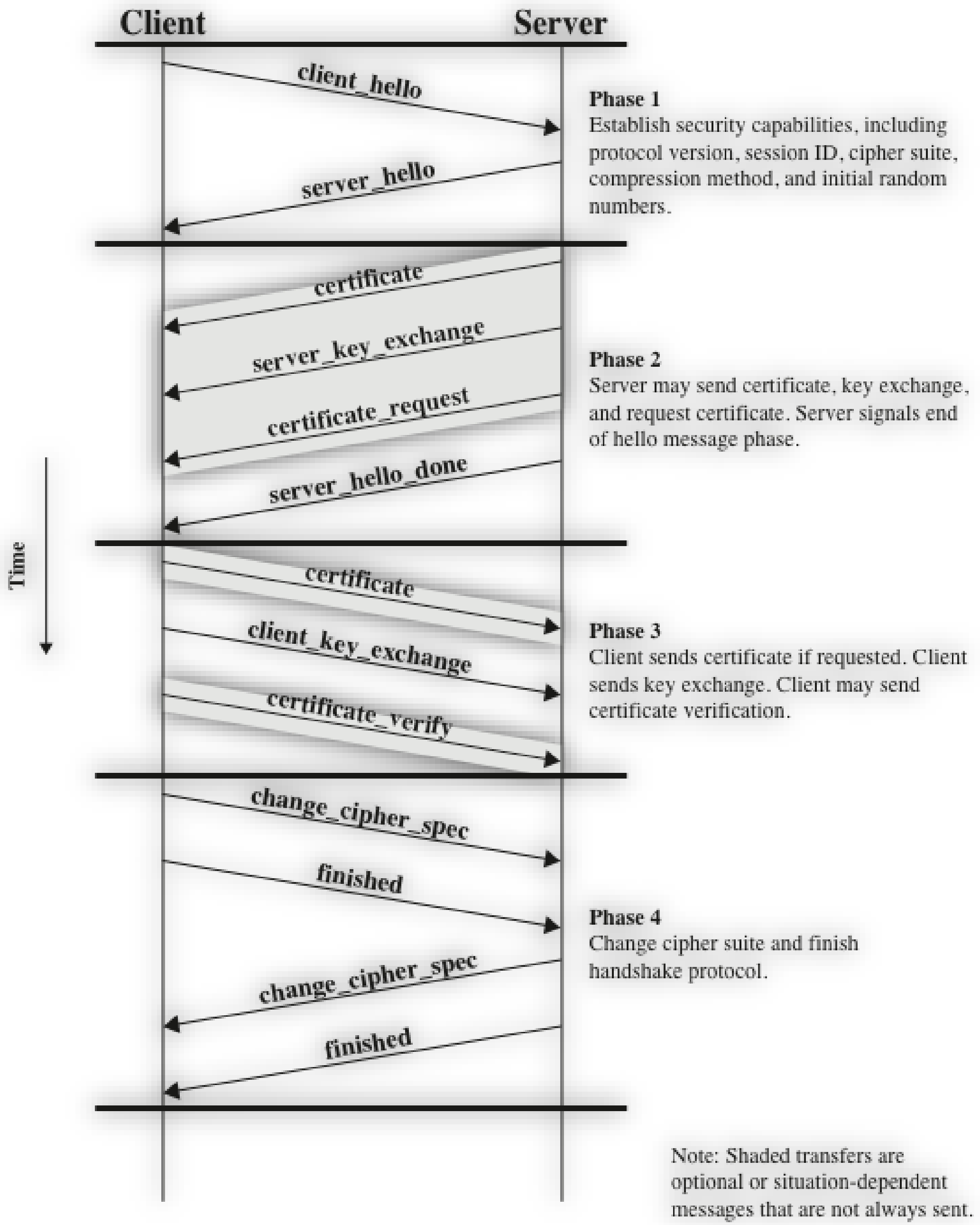
Handshake Payload and Types



1 byte	3 bytes	0 bytes
Type	Length	Content



Message Type	Parameters
hello_request	null
client_hello	version, random, session id, cipher suite, compression method
server_hello	version, random, session id, cipher suite, compression method
certificate	chain of X.509v3 certificates
server_key_exchange	parameters, signature
certificate_request	type, authorities
server_done	null
certificate_verify	signature
client_key_exchange	parameters, signature
finished	hash value



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.

Phase 3
Client sends certificate if requested. Client sends key exchange. Client may send certificate verification.

Phase 4
Change cipher suite and finish handshake protocol.

Note: Shaded transfers are optional or situation-dependent messages that are not always sent.

Figure 6.6 Handshake Protocol Action

SSL – نتیجه گیری

- SSL نیازهای امنیتی زیر را فراهم می کند
  - محرمانگی
    - رمزنگاری متقارن
  - صحت داده
    - کد احراز هویت داده
  - احراز هویت
    - استاندارد x.509
- امروزه مهمترین کاربرد SSL در قرارداد HTTPS می باشد.

TLS:
Transport Layer Security

- یک استاندارد از IETF
- به دنبال ایجاد یک نسخه استاندارد اینترنتی از SSL می باشد
- بسیار شبیه SSL نسخه ۳ با تغییرات جزئی
- در حال حاضر نسخه های TLS 1.0, 1.1, 1.2, 1.3
- طی توافقی از March 2020 نسخه‌های 1.0, 1.1 توسط مایکروسافت، اپل، موزیلا و گوگل به طور کامل کنار گذارده خواهند شد.
  - □ کنار گذاردن الگوریتم‌ها و قابلیتهای ضعیف مانند MD5, SHA-1
  - □ استفاده از الگوریتم‌های جدید

Transport Layer Security (TLS)

- An IETF standardization initiative whose goal is to produce an Internet standard version of SSL
- Is defined as a Proposed Internet Standard in RFC 5246
  - RFC 5246 is very similar to SSLv3

Differences include:

- Version number
- Message Authentication Code
- Pseudorandom function
- Alert keys
- Cipher suites
- Client certificate types
- Certificate_verify and Finished Messages
- Cryptographic computations
- Padding

HTTPS
(HTTP over SSL)

- Refers to the combination of HTTP and SSL to implement secure communication between a Web browser and a Web server
- The HTTPS capability is built into all modern Web browsers
- A user of a Web browser will see URL addresses that begin with https:// rather than http://
- If HTTPS is specified, port 443 is used, which invokes SSL
- Documented in RFC 2818, *HTTP Over TLS*
  - There is no fundamental change in using HTTP over either SSL or TLS and both implementations are referred to as HTTPS
- When HTTPS is used, the following elements of the communication are encrypted:
  - URL of the requested document
  - Contents of the document
  - Contents of browser forms
  - Cookies sent from browser to server and from server to browser
  - Contents of HTTP header

Connection Initiation

For HTTPS, the agent acting as the HTTP client also acts as the TLS client

- The client initiates a connection to the server on the appropriate port and then sends the TLS ClientHello to begin the TLS handshake
- When the TLS handshake has finished, the client may then initiate the first HTTP request
- All HTTP data is to be sent as TLS application data

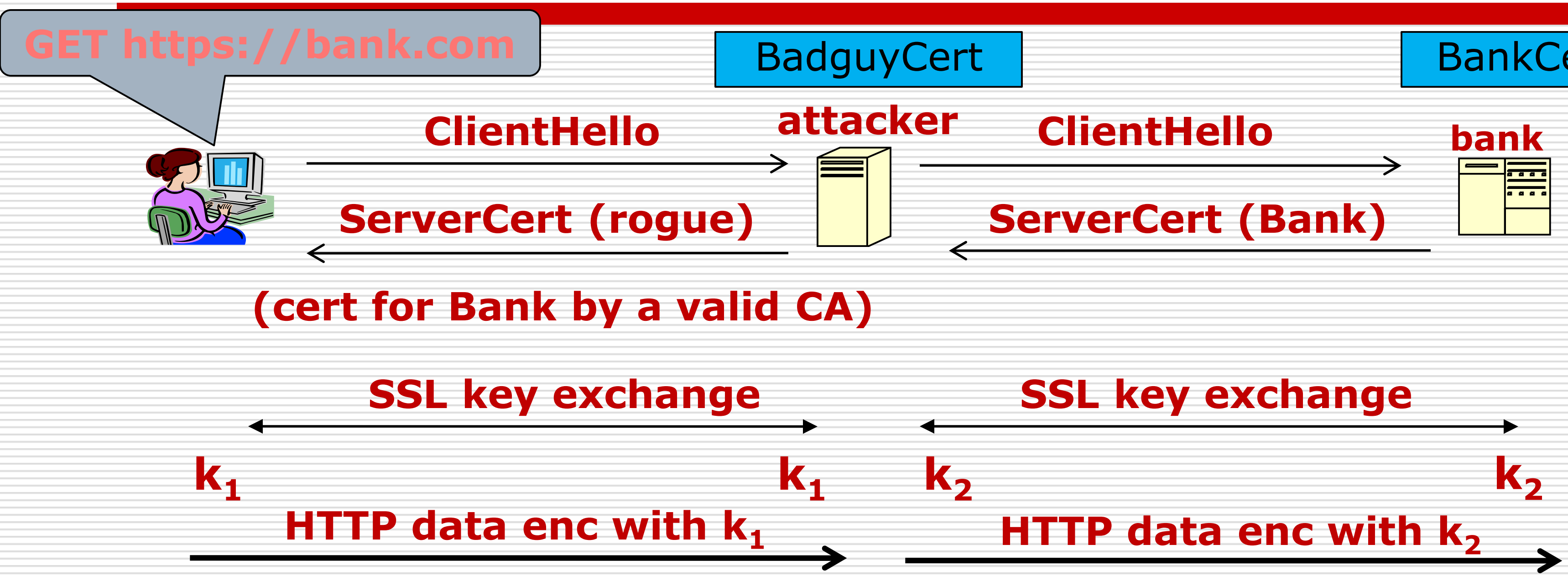
There are three levels of awareness of a connection in HTTPS:

- At the HTTP level, an HTTP client requests a connection to an HTTP server by sending a connection request to the next lowest layer
  - Typically the next lowest layer is TCP, but it may also be TLS/SSL
- At the level of TLS, a session is established between a TLS client and a TLS server
  - This session can support one or more connections at any time
- A TLS request to establish a connection begins with the establishment of a TCP connection between the TCP entity on the client side and the TCP entity on the server side

Connection Closure

- An HTTP client or server can indicate the closing of a connection by including the line `Connection: close` in an HTTP record
- The closure of an HTTPS connection requires that TLS close the connection with the peer TLS entity on the remote side, which will involve closing the underlying TCP connection
- TLS implementations must initiate an exchange of closure alerts before closing a connection
  - A TLS implementation may, after sending a closure alert, close the connection without waiting for the peer to send its closure alert, generating an "incomplete close"
- An unannounced TCP closure could be evidence of some sort of attack so the HTTPS client should issue some sort of security warning when this occurs

Man in the middle attack using rogue cert



GET https://bank.com

BadguyCert

BankCert

ClientHello

attacker

ClientHello

bank

ServerCert (rogue)

ServerCert (Bank)

(cert for Bank by a valid CA)

SSL key exchange

SSL key exchange

k_1 k_1 k_2 k_2

HTTP data enc with k_1

HTTP data enc with k_2

Attacker proxies data between user and bank.
Sees all traffic and can modify data at will.

What to do? (many good ideas)

1. **Public-key pinning (static pins)**
   - Hardcode list of allowed CAs for certain sites (Gmail, facebook, …)
   - Browser rejects certs issued by a CA not on list
   - Now deprecated (because often incorrectly used in practice)

1. **Certificate Transparency (CT)**: [LL'12]
   - idea: CA's must advertise a log of all certs. they issued
   - Browser will only use a cert if it is published on (two) log servers
     - Server attaches a signed statement from log (SCT) to certificate
   - Companies can scan logs to look for invalid issuance

CT requirements

April 30, 2018:   CT required by chrome

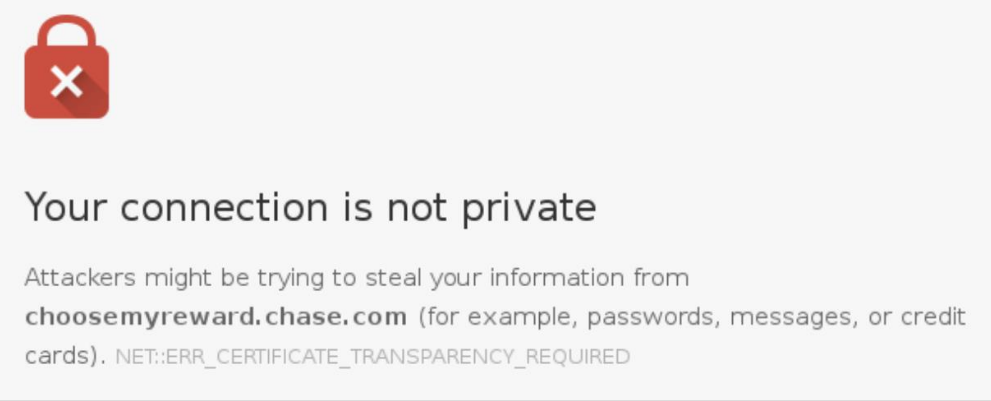
- Required for all certificates with a path to a trusted root CA

    (not required for an installed root CA)

- Otherwise: HTTPS errors

Cert for crypto.stanford.edu published on five logs:

cloudflare_nimbus2018
google_argon2018,
google_aviator
google_pilot,   google_rocketeer



Your connection is not private

Attackers might be trying to steal your information from **choosemyreward.chase.com** (for example, passwords, messages, or credit cards). NET::ERR_CERTIFICATE_TRANSPARENCY_REQUIRED

3. Mixed Content: HTTP and HTTPS

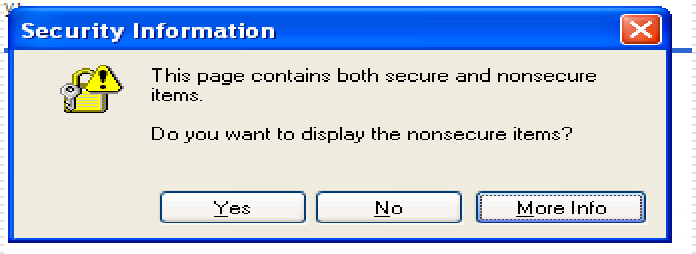
Page loads over HTTPS, but contains content over HTTP

(e.g. <script src="http://.../script.js>)

never write this

⇒ Active network attacker can hijack session by modifying script en-route to browser

IE7:

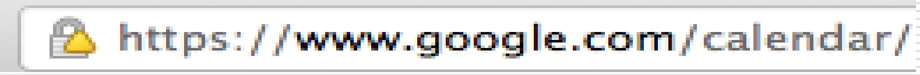


Security Information

This page contains both secure and nonsecure items.

Do you want to display the nonsecure items?

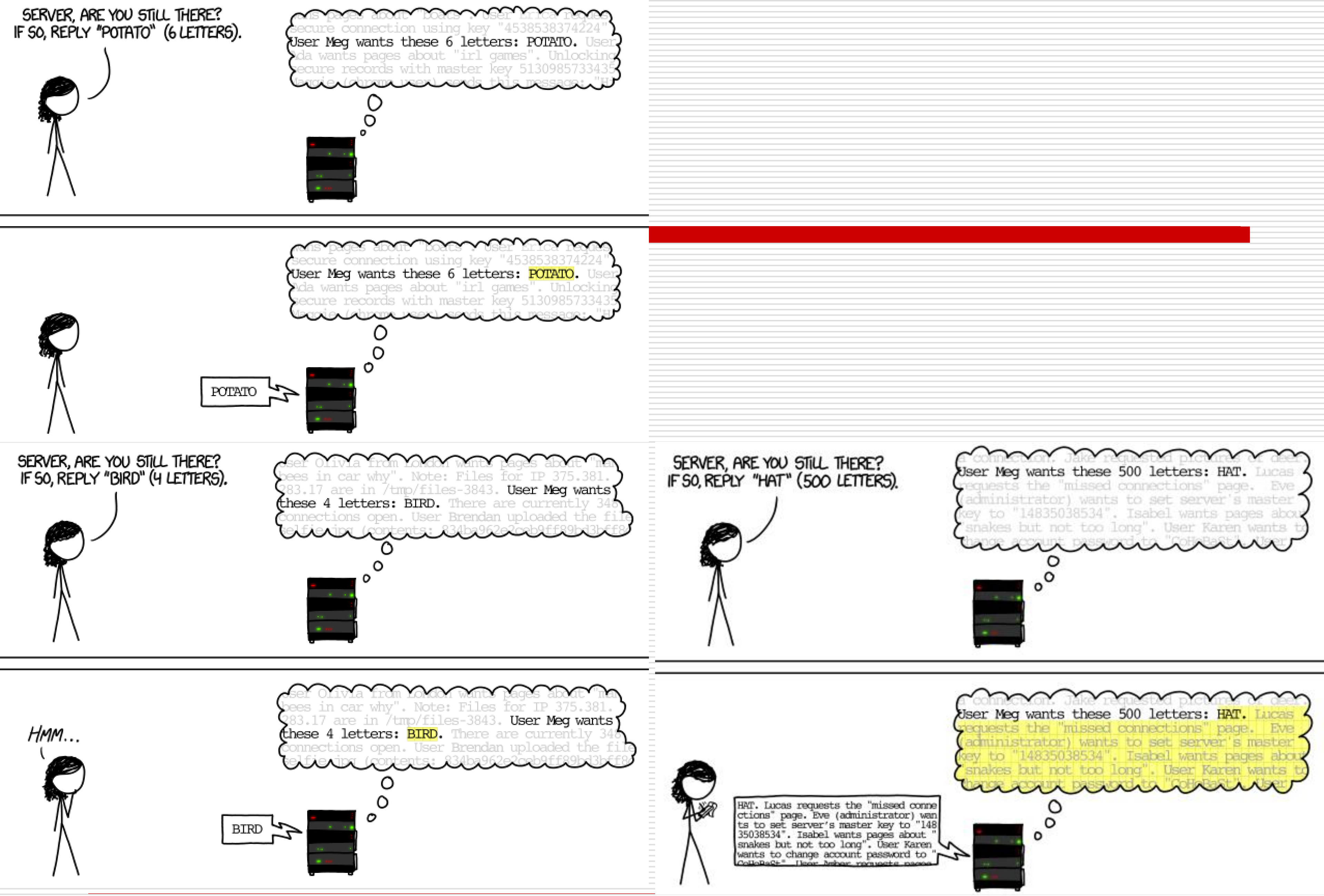
Old Chrome:



https://www.google.com/calendar/

Mostly ignored by users …

OpenSSL Heartbleed Attack


SERVER, ARE YOU STILL THERE? IF SO, REPLY "POTATO" (6 LETTERS).
ants pages about "boats". User Erica requests secure connection using key "4538538374224". User Meg wants these 6 letters: POTATO. User Ada wants pages about "irl games". Unlocking secure records with master key 5130985733435. Maggie (chrome user) sends this message: "H
POTATO
SERVER, ARE YOU STILL THERE? IF SO, REPLY "BIRD" (4 LETTERS).
User Olivia from London wants pages about "ma bees in car why". Note: Files for IP 375.381.283.17 are in /tmp/files-3843. User Meg wants these 4 letters: BIRD. There are currently 346 connections open. User Brendan uploaded the file selfie.jpg (contents: 834ba962e2ceb9ff89bd3bff8
HMM...
BIRD
SERVER, ARE YOU STILL THERE? IF SO, REPLY "HAT" (500 LETTERS).
a connection. Jake requested pictures of deer. User Meg wants these 500 letters: HAT. Lucas requests the "missed connections" page. Eve (administrator) wants to set server's master key to "14835038534". Isabel wants pages about "snakes but not too long". User Karen wants to change account password to "CoHoBaSt". User
HAT. Lucas requests the "missed connections" page. Eve (administrator) wants to set server's master key to "14835038534". Isabel wants pages about "snakes but not too long". User Karen wants to change account password to "CoHoBaSt". User Amber requests pages

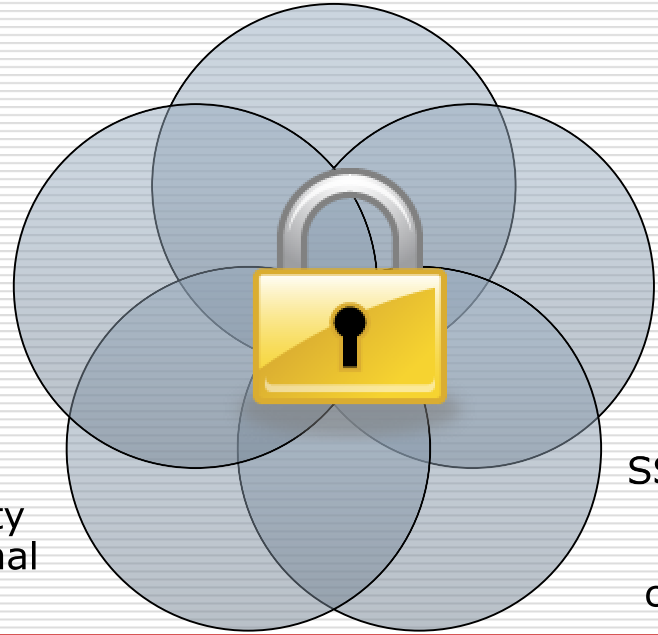
SECURE SHELL (SSH)

Secure Shell (SSH)

A protocol for secure network communications designed to be relatively simple and inexpensive to implement

SSH client and server applications are widely available for most operating systems

- Has become the method of choice for remote login and X tunneling
- Is rapidly becoming one of the most pervasive applications for encryption technology outside of embedded systems

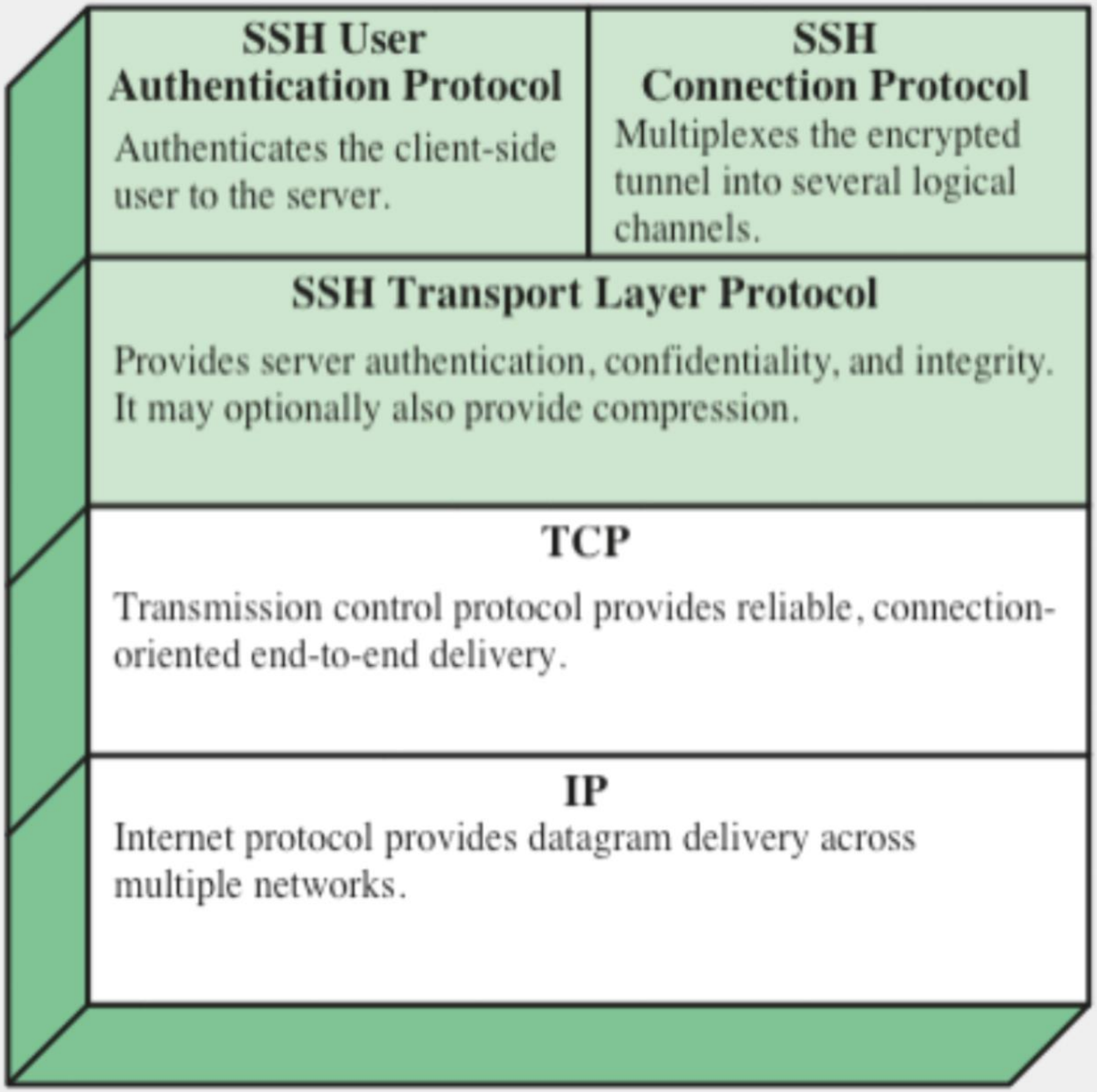


The initial version, SSH1 was focused on providing a secure remote logon facility to replace TELNET and other remote logon schemes that provided no security

SSH2 fixes a number of security flaws in the original scheme

- Is documented as a proposed standard in IETF RFCs 4250 through 4256

SSH also provides a more general client/server capability and can be used for such network functions as file transfer and e-mail



SSH User Authentication Protocol

Authenticates the client-side user to the server.

SSH Connection Protocol

Multiplexes the encrypted tunnel into several logical channels.

SSH Transport Layer Protocol

Provides server authentication, confidentiality, and integrity. It may optionally also provide compression.

TCP

Transmission control protocol provides reliable, connection-oriented end-to-end delivery.

IP

Internet protocol provides datagram delivery across multiple networks.

Figure 6.8 SSH Protocol Stack

Transport Layer Protocol

- Server authentication occurs at the transport layer, based on the server possessing a public/private key pair
- A server may have multiple host keys using multiple different asymmetric encryption algorithms
- Multiple hosts may share the same host key
- The server host key is used during key exchange to authenticate the identity of the host
- RFC 4251 dictates two alternative trust models:
  - The client has a local database that associates each host name with the corresponding public host key
  - The host name-to-key association is certified by a trusted certification authority (CA); the client only knows the CA root key and can verify the validity of all host keys certified by accepted CAs

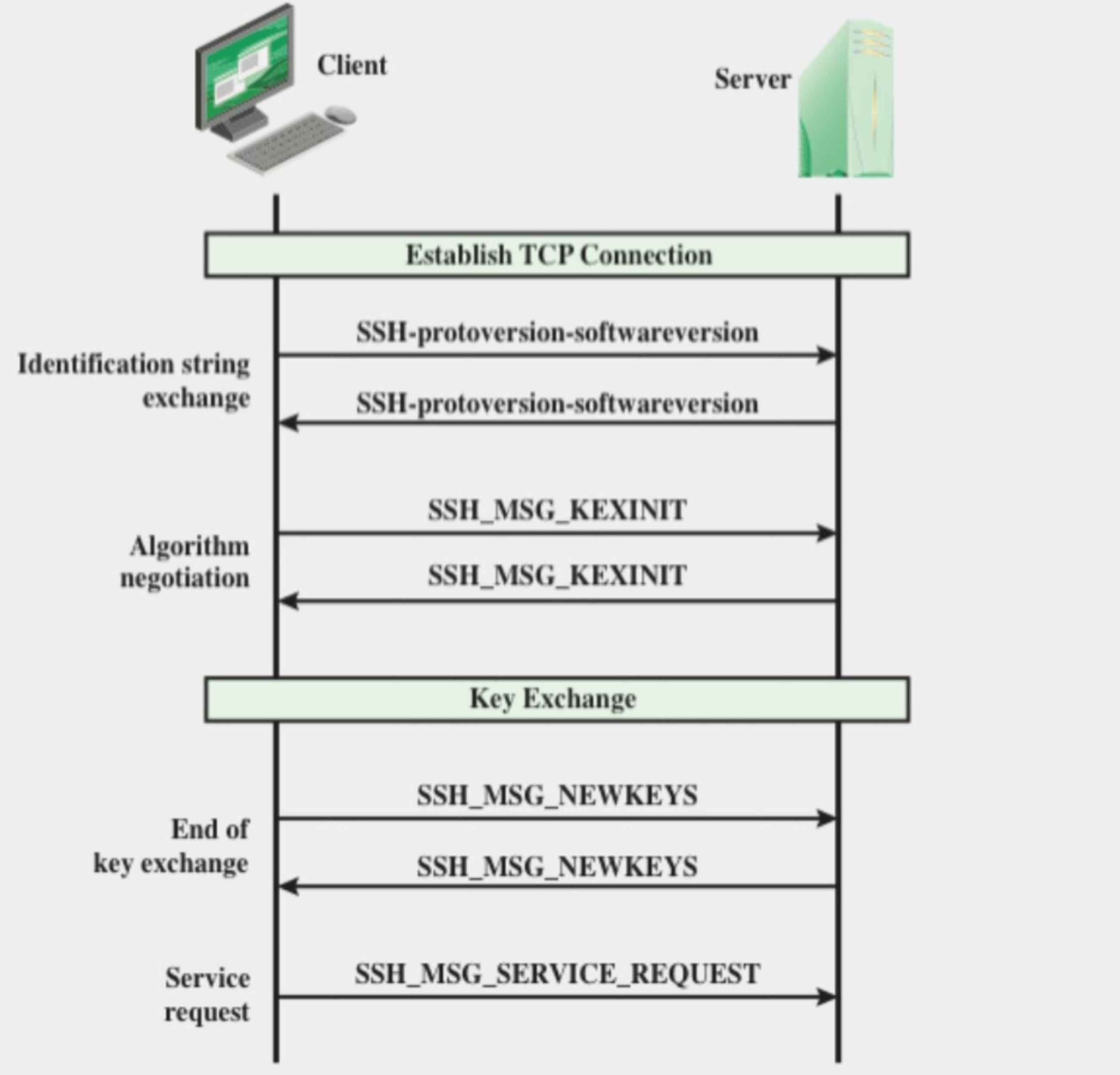
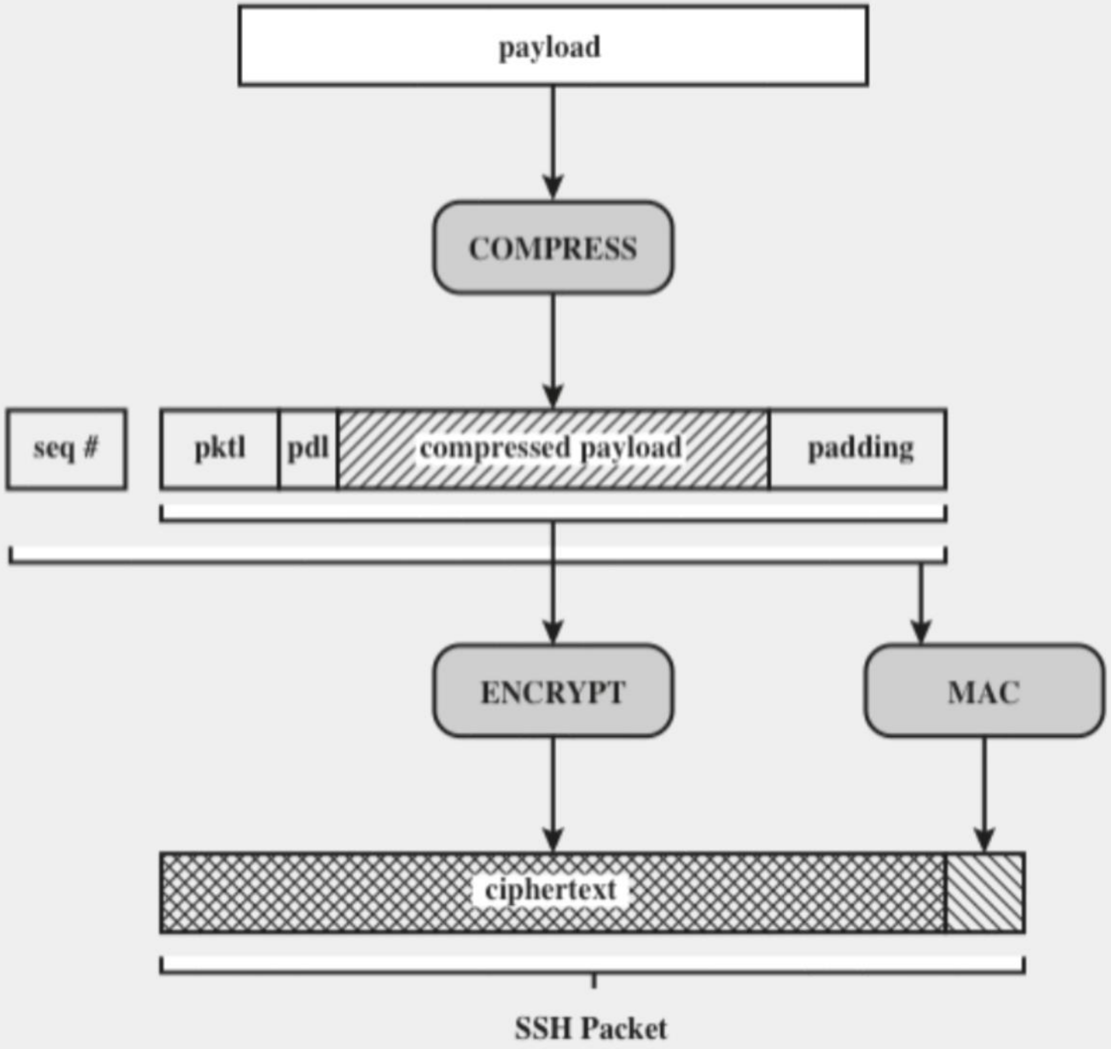


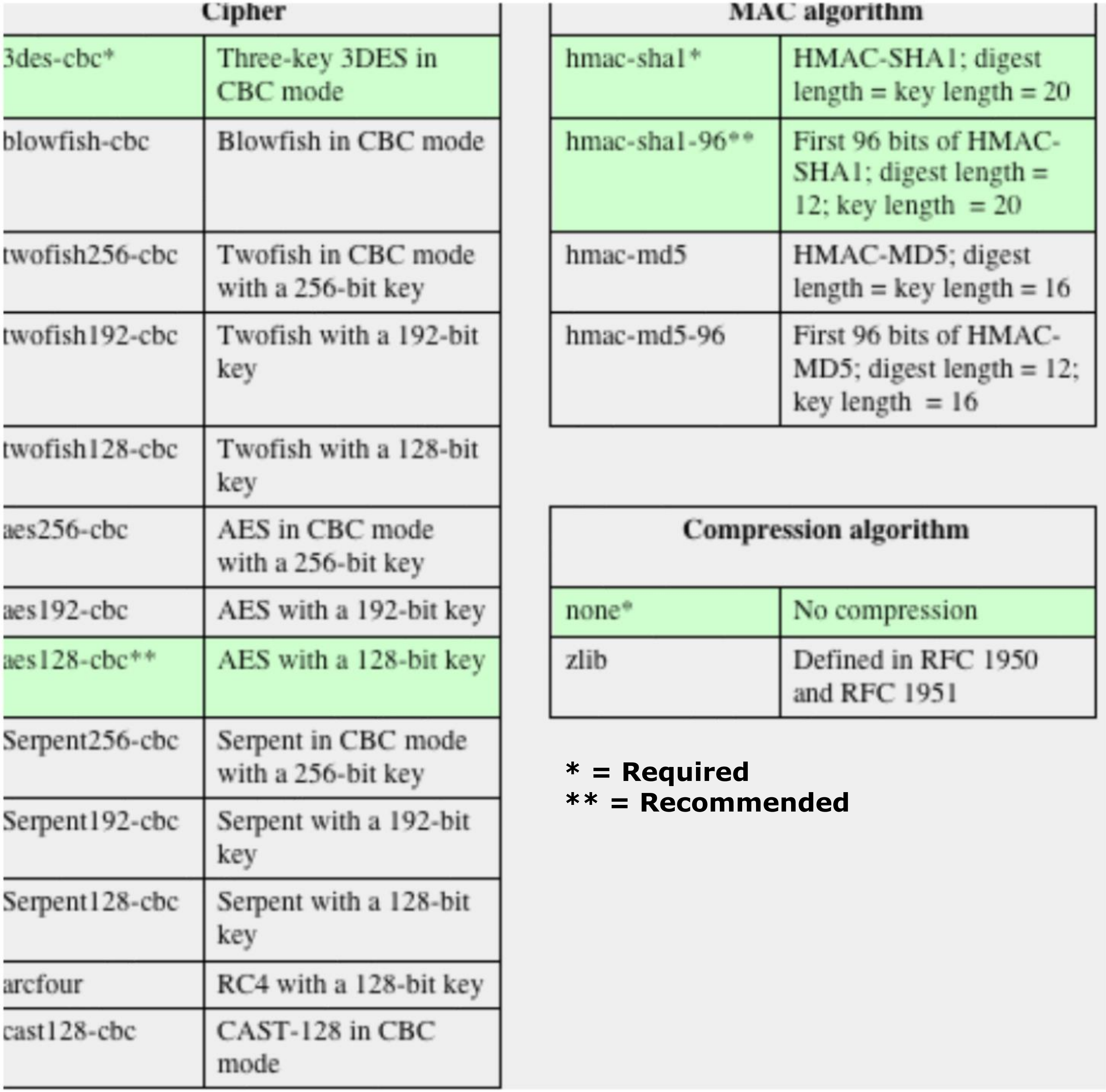
Figure 6.9 SSH Transport Layer Protocol Packet Exchanges



pktl = packet length
pdl = padding length

Figure 6.10 SSH Transport Layer Protocol Packet Formation

Table 6.3 SSH Transport Layer Cryptographic Algorithms



Cipher	
3des-cbc*	Three-key 3DES in CBC mode
blowfish-cbc	Blowfish in CBC mode
twofish256-cbc	Twofish in CBC mode with a 256-bit key
twofish192-cbc	Twofish with a 192-bit key
twofish128-cbc	Twofish with a 128-bit key
aes256-cbc	AES in CBC mode with a 256-bit key
aes192-cbc	AES with a 192-bit key
aes128-cbc**	AES with a 128-bit key
Serpent256-cbc	Serpent in CBC mode with a 256-bit key
Serpent192-cbc	Serpent with a 192-bit key
Serpent128-cbc	Serpent with a 128-bit key
arcfour	RC4 with a 128-bit key
cast128-cbc	CAST-128 in CBC mode

MAC algorithm	
hmac-sha1*	HMAC-SHA1; digest length = key length = 20
hmac-sha1-96**	First 96 bits of HMAC-SHA1; digest length = 12; key length = 20
hmac-md5	HMAC-MD5; digest length = key length = 16
hmac-md5-96	First 96 bits of HMAC-MD5; digest length = 12; key length = 16

Compression algorithm	
none*	No compression
zlib	Defined in RFC 1950 and RFC 1951

* = Required
** = Recommended

Authentication Methods

Publickey

- The client sends a message to the server that contains the client’s public key, with the message signed by the client’s private key
- When the server receives this message, it checks whether the supplied key is acceptable for authentication and, if so, it checks whether the signature is correct

Password

- The client sends a message containing a plaintext password, which is protected by encryption by the Transport Layer Protocol

Hostbased

- Authentication is performed on the client’s host rather than the client itself
- This method works by having the client send a signature created with the private key of the client host
- Rather than directly verifying the user’s identity, the SSH server verifies the identity of the client host

Connection Protocol

- The SSH Connection Protocol runs on top of the SSH Transport Layer Protocol and assumes that a secure authentication connection is in use
  - The secure authentication connection, referred to as a *tunnel,* is used by the Connection Protocol to multiplex a number of logical channels

- Channel mechanism
  - All types of communication using SSH are supported using separate channels
  - Either side may open a channel
  - For each channel, each side associates a unique channel number
  - Channels are flow controlled using a window mechanism
  - No data may be sent to a channel until a message is received to indicate that window space is available
  - The life of a channel progresses through three stages: opening a channel, data transfer, and closing a channel

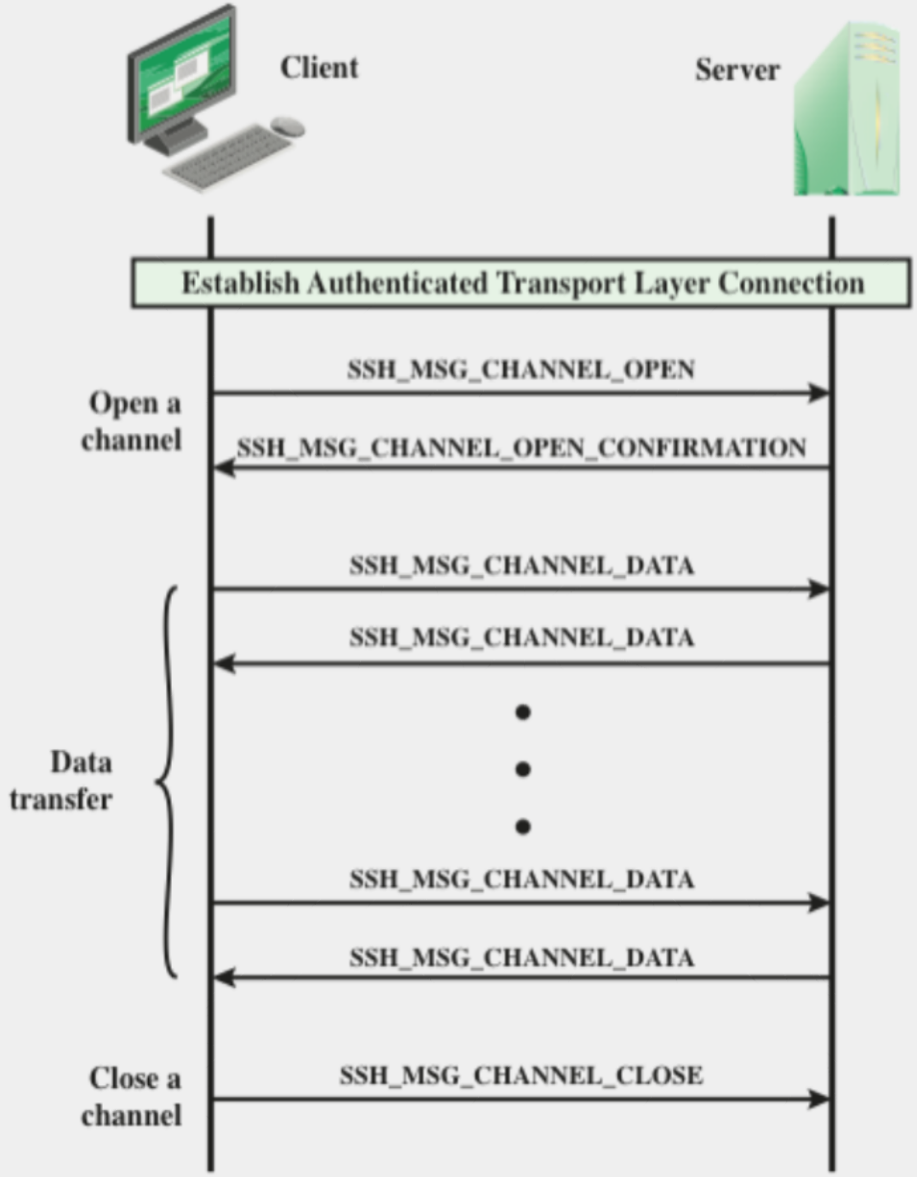


Figure 6.11 Example SSH Connection Protocol Message Exchange

Channel Types

Four channel types are recognized in the SSH Connection Protocol specification

Session

- The remote execution of a program
- The program may be a shell, an application such as file transfer or e-mail, a system command, or some built-in subsystem
- Once a session channel is opened, subsequent requests are used to start the remote program

X11

- Refers to the X Window System, a computer software system and network protocol that provides a graphical user interface (GUI) for networked computers
- X allows applications to run on a network server but to be displayed on a desktop machine

Forwarded-tcpip

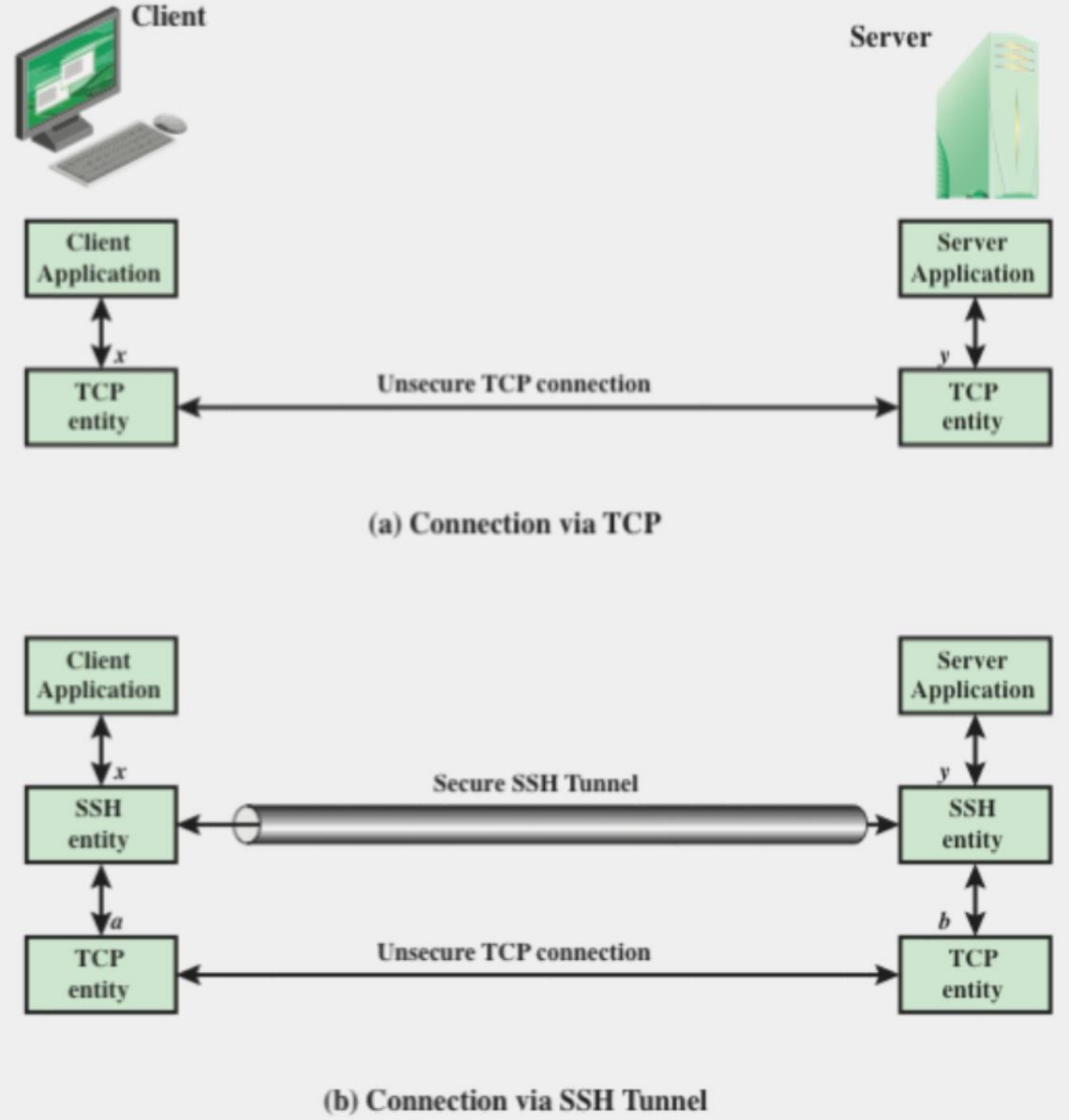
- Remote port forwarding

Direct-tcpip

- Local port forwarding

Port Forwarding

- One of the most useful features of SSH
- Provides the ability to convert any insecure TCP connection into a secure SSH connection (also referred to as SSH tunneling)
- Incoming TCP traffic is delivered to the appropriate application on the basis of the port number (a port is an identifier of a user of TCP)
- An application may employ multiple port numbers



(a) Connection via TCP

(b) Connection via SSH Tunnel

Figure 6.12 SSH Transport Layer Packet Exchanges

آینده امنیت لایه انتقال ...

- □ جایگزینی پروتکل TCP با یک پروتکل امن!
- □ پروتکل QUIC
  - ■ طراحی و پیشنهاد شده توسط google در سال ۲۰۱۲
  - ■ هدف: کاهش سربار برقراری اتصال (در HTTPS دو تا لازم است یکی برای TCP و یکی هم TLS)
  - ■ هنوز فراگیر نشده است
  - ■ پشتیبانی توسط مرورگرهای کروم و کرومیوم، Opera
  - ■ پشتیبانی توسط سرورهای گوگل
  - ■ در سال ۲۰۱۸ گروه کاری HTTP و QUIC پروتکل جدیدی برای وب پیشنهاد دادند:
    - ■ HTTP/3 or H3= HTTP + QUIC

برای مطالعه بیشتر ...

- □ IETF main page: https://www.ietf.org
- □ TLS WG: https://tlswg.org/
- □ QUIC WG: https://quicwg.org/
- □ DOH WG: https://datatracker.ietf.org/wg/doh/about/

یا ذالامن و الامان

پایان

؟