

یا ذالامن والامان

امنیت لایه انتقال: SSL/TLS, SSH

مبتنی بر فصل ۶ از کتاب

Network Security Essentials

ویرایش شده توسط: حمید رضا شهریاری

<http://ceit.aut.ac.ir/~shahriari>

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	<ul style="list-style-type: none"> •Modification of user data •Trojan horse browser •Modification of memory •Modification of message traffic in transit 	<ul style="list-style-type: none"> •Loss of information •Compromise of machine •Vulnerability to all other threats 	Cryptographic checksums
Confidentiality	<ul style="list-style-type: none"> •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 	<ul style="list-style-type: none"> •Loss of information •Loss of privacy 	Encryption, Web proxies
Denial of Service	<ul style="list-style-type: none"> •Killing of user threads •Flooding machine with bogus requests •Filling up disk or memory •Isolating machine by DNS attacks 	<ul style="list-style-type: none"> •Disruptive •Annoying •Prevent user from getting work done 	Difficult to prevent
Authentication	<ul style="list-style-type: none"> •Impersonation of legitimate users •Data forgery 	<ul style="list-style-type: none"> •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

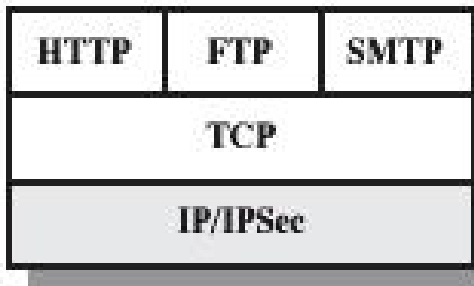
■ شفاف از دید برنامه‌های کاربردی سطح بالاتر

■ پشتیبانی مرورگرها متداول و نیز بسیاری از وب سرورها

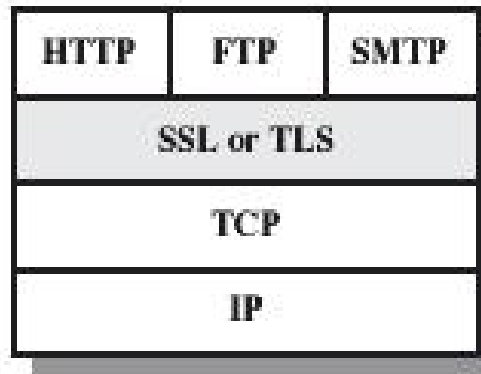
□ سرویسهای امنیتی وابسته به کاربرد خاص

■ مانند PGP

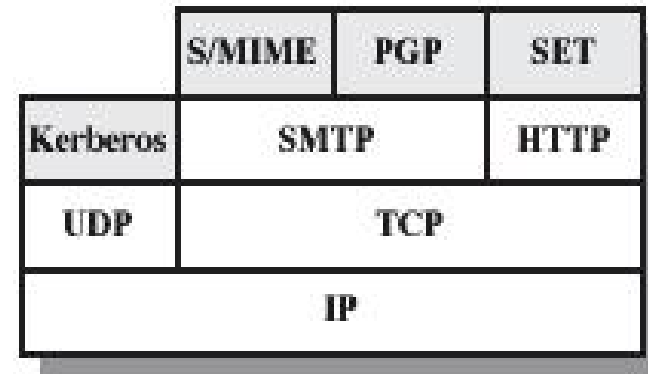
Web Security Approaches



(a) Network Level



(b) Transport Level



(c) Application Level

SSL – تاریخچه

July, 1994 ☐

- شرکت Netscape طراحی SSL 1.0 را انجام داد.
- این نسخه هیچگاه منتشر نشد!

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 ۴۰ بیتی در خارج آمریکا

SSL – تاریخچه (...ادامه)

July, 1995 ☐

- مایکروسافت نسخه جدیدی از IE را به بازار عرضه کرد که از SSL پشتیبانی می کرد.
- پشتیبانی از مدهای کاری جدید و افزایش طول کلیدهای قابل استفاده

Nov, 1995 ☐

- شرکت Netscape توصیف SSL 3.0 را منتشر کرد
- با تغییرات و جهش عمده نسبت به نسخه های قبلی همراه بود
- ضمن اینکه نسبت به نسخه SSL v2.0، Backward Compatible بود

SSL – تاریخچه (...ادامه)

May, 1996 ☐

■ IETF گروه کاری TLS را تشکیل داد و مسئولیت پاسخگویی به مشکلات قرارداد SSL را برعهده گرفت.

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.

SSL

- لایه امنیتی در بالای لایه انتقال
- ارائه شده توسط شرکت Netscape و نسخه ۳ آن نسخه استاندارد اینترنت می باشد
- سرویس قابل اطمینان انتها به انتها (end to end) و مبتنی بر TCP
- پروتکل آن در دو لایه پیاده سازی می شود

SSL – معماری

□ دو مفهوم اساسی در SSL:

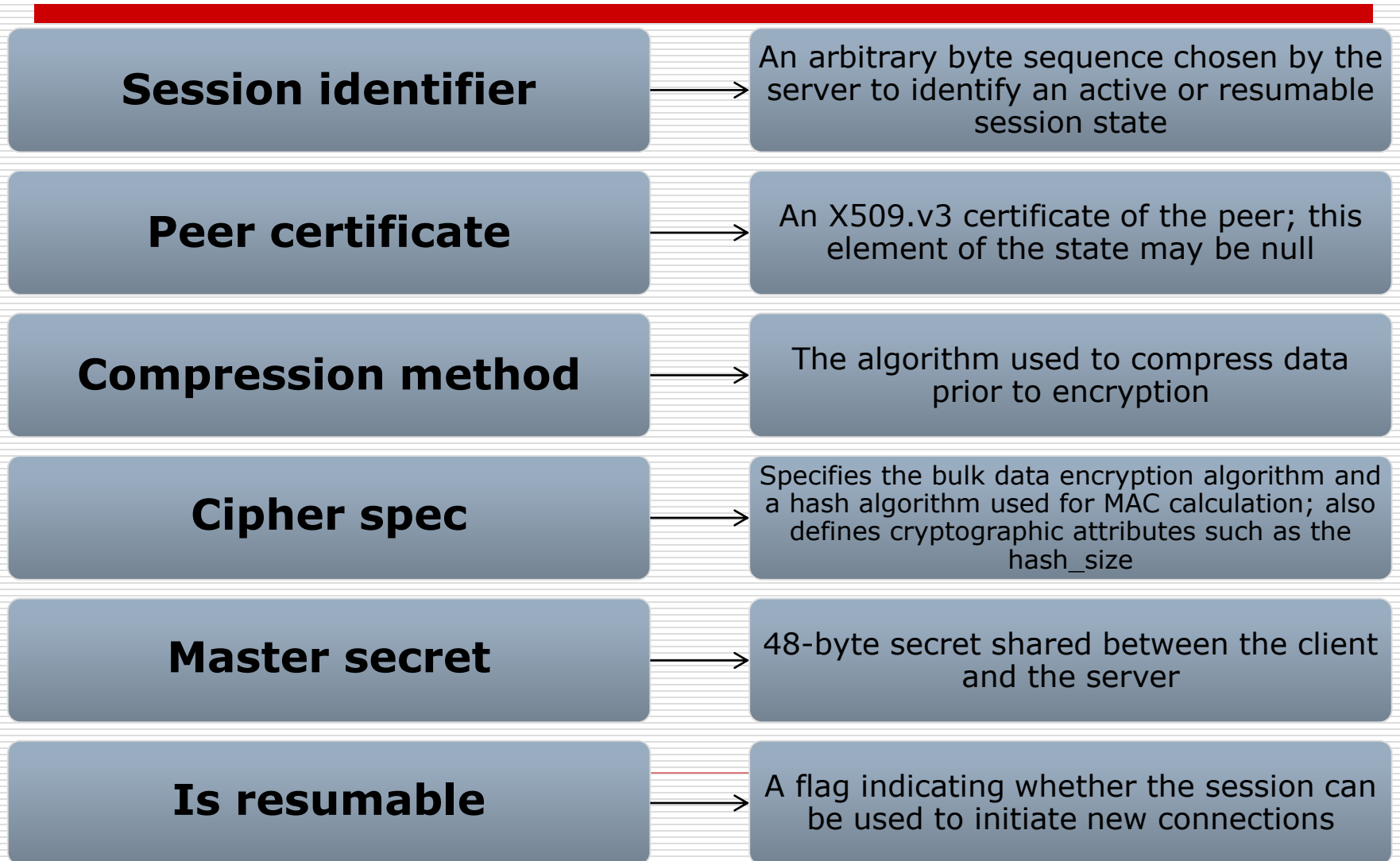
SSL connection

- 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 با پارامترهای زیر مشخص می‌شود



یک حالت اتصال با پارمترهای زیر مشخص می شود

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

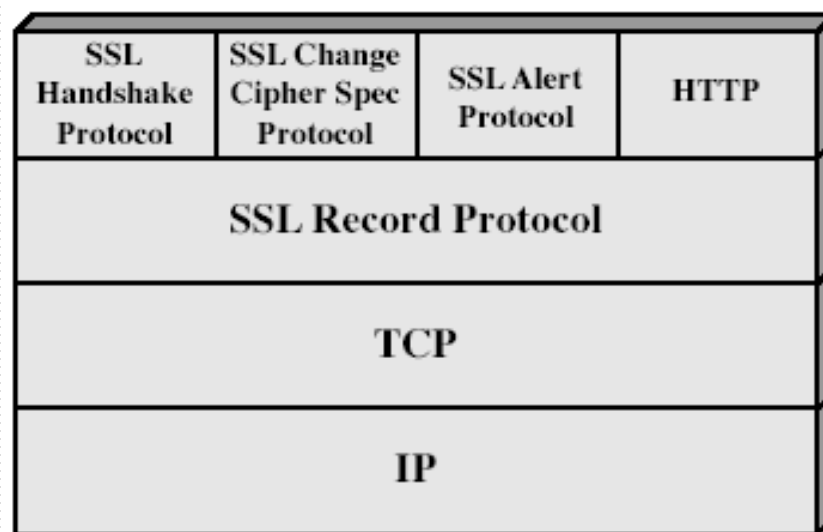
- When a block cipher in CBC mode is used, an initialization vector (IV) is maintained for each key
- This field is first initialized by the SSL Handshake Protocol
- The final ciphertext block from each record is preserved for use as the IV with the following record

Sequence
numbers

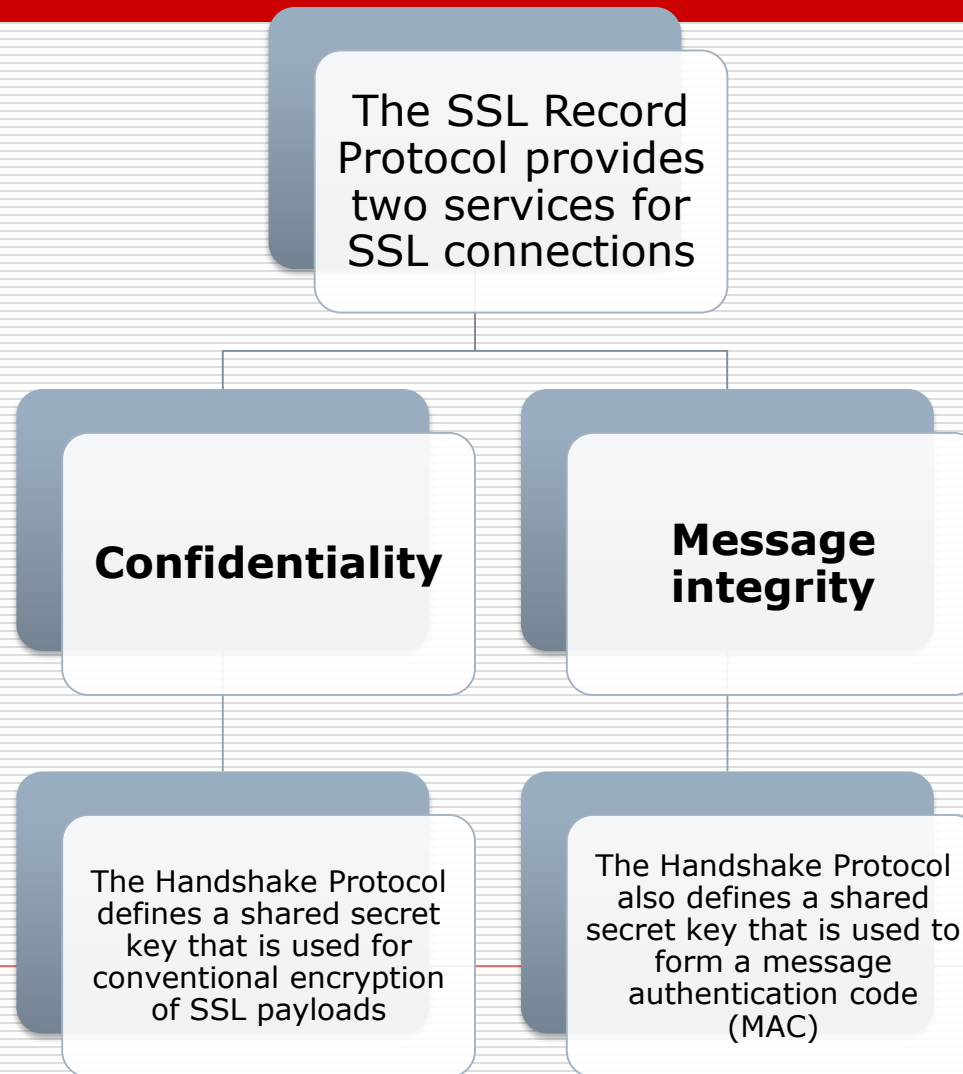
- Each party maintains separate sequence numbers for transmitted and received messages for each connection
- When a party sends or receives a change cipher spec message, the appropriate sequence number is set to zero
- Sequence numbers may not exceed $2^{64} - 1$

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 Record Protocol



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

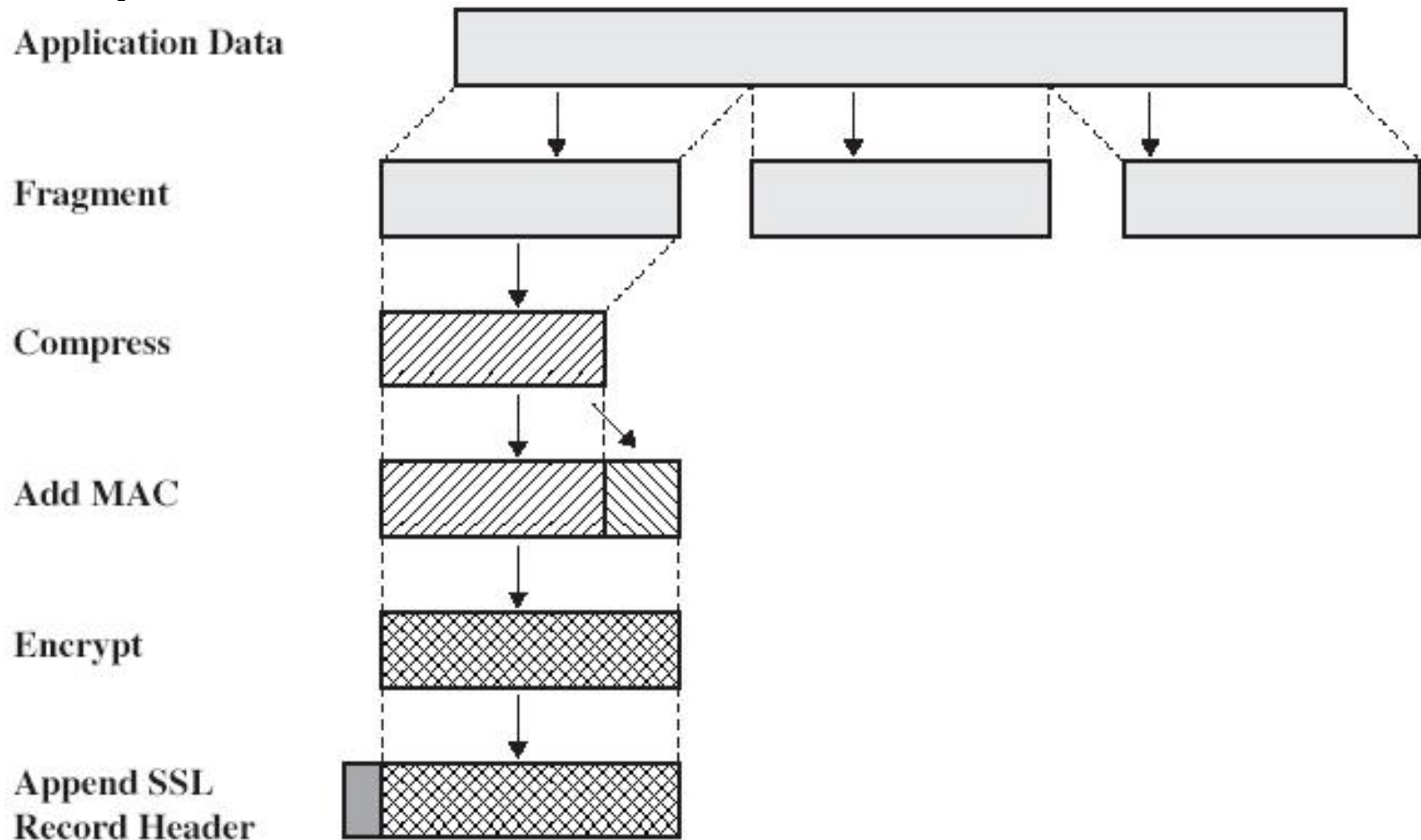
□ صحت پیغام

□ 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

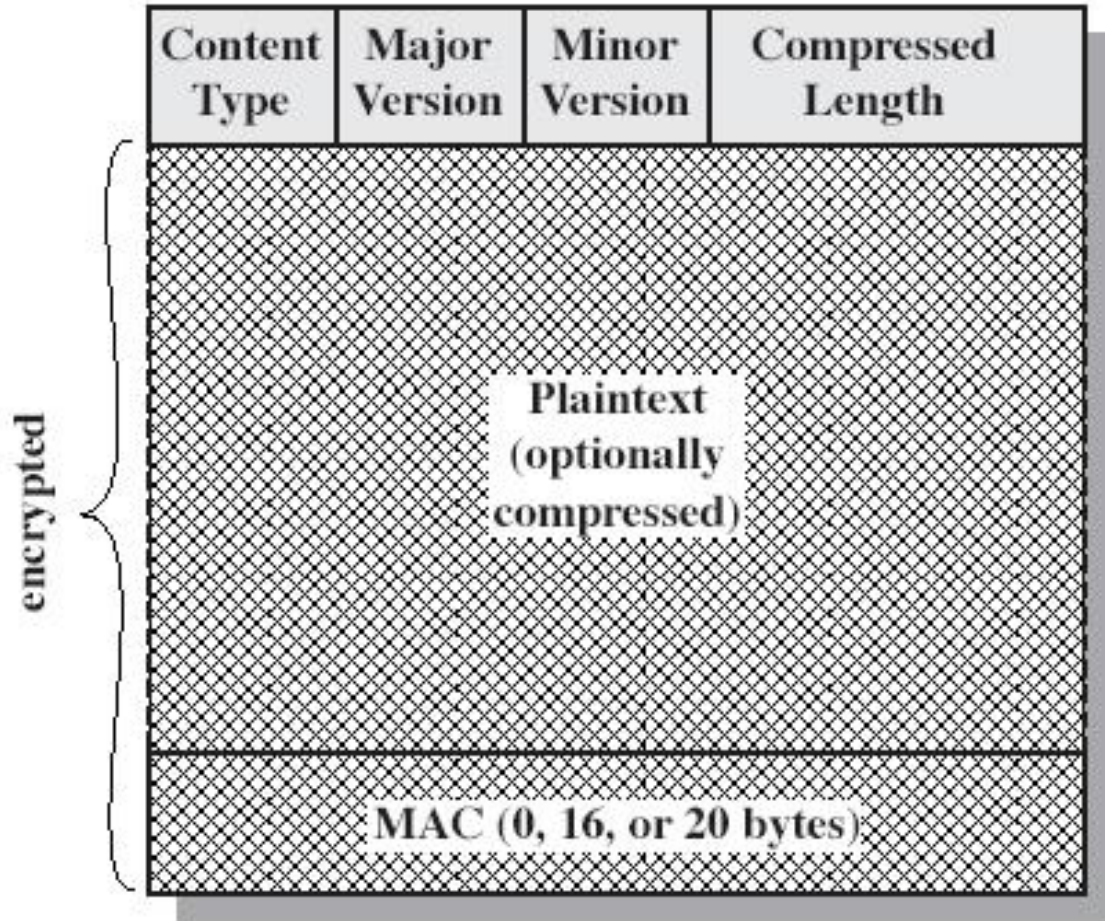


SSL – پروتکل‌ها

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 نهی.
- **اضافه کردن سرآیند:** به ابتدای بلاک رمز شده 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



SSL – پروتکلها

پروتکل Change Cipher Spec:

- یکی از ۳ پروتکل لایه دوم SSL که از پروتکل Record استفاده می کنند.
- شامل ۱ بایت می باشد
- منجر به نوشته شدن مشخصات رمزنگاری معلق (pending) بجای مشخصات فعلی می شود.

1 byte

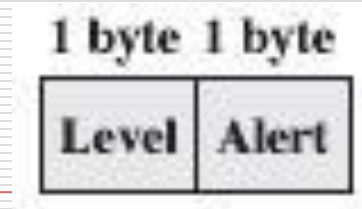
1

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



SSL – پروتکلها

SSL Handshake پروتکل

- 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 ۴ 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 پشتیبانی شده توسط کارفرما
- پیشنهاد الگوریتم های مناسب و روش تبادل کلید آنها
- پیشنهاد مکانیسم فشرده سازی مناسب
- انتخاب نسخه و الگوریتم های مورد قبول کارگزار
- کارگزار بررسی می کند 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 سری

■ کارفرما کلید سری را تولید کرده و برای کارگزار می فرستد

■ یا این که هر دو با استفاده از پارامترهای DH کلید سری را محاسبه می کنند.

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

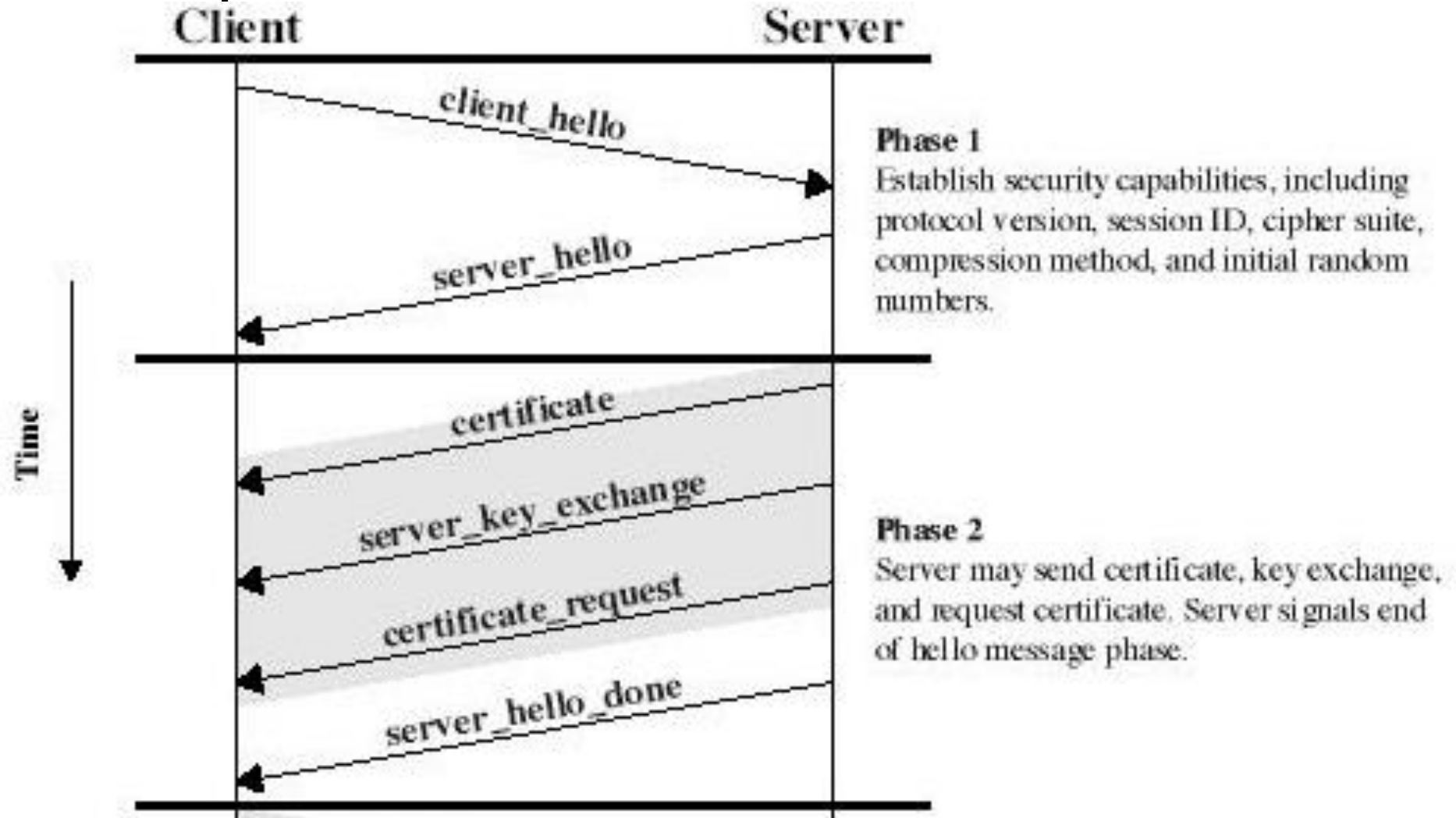
- ارسال پیغام پایانی
- آغاز تبادل اطلاعات به صورت محرمانه و با پارامترهای جدید

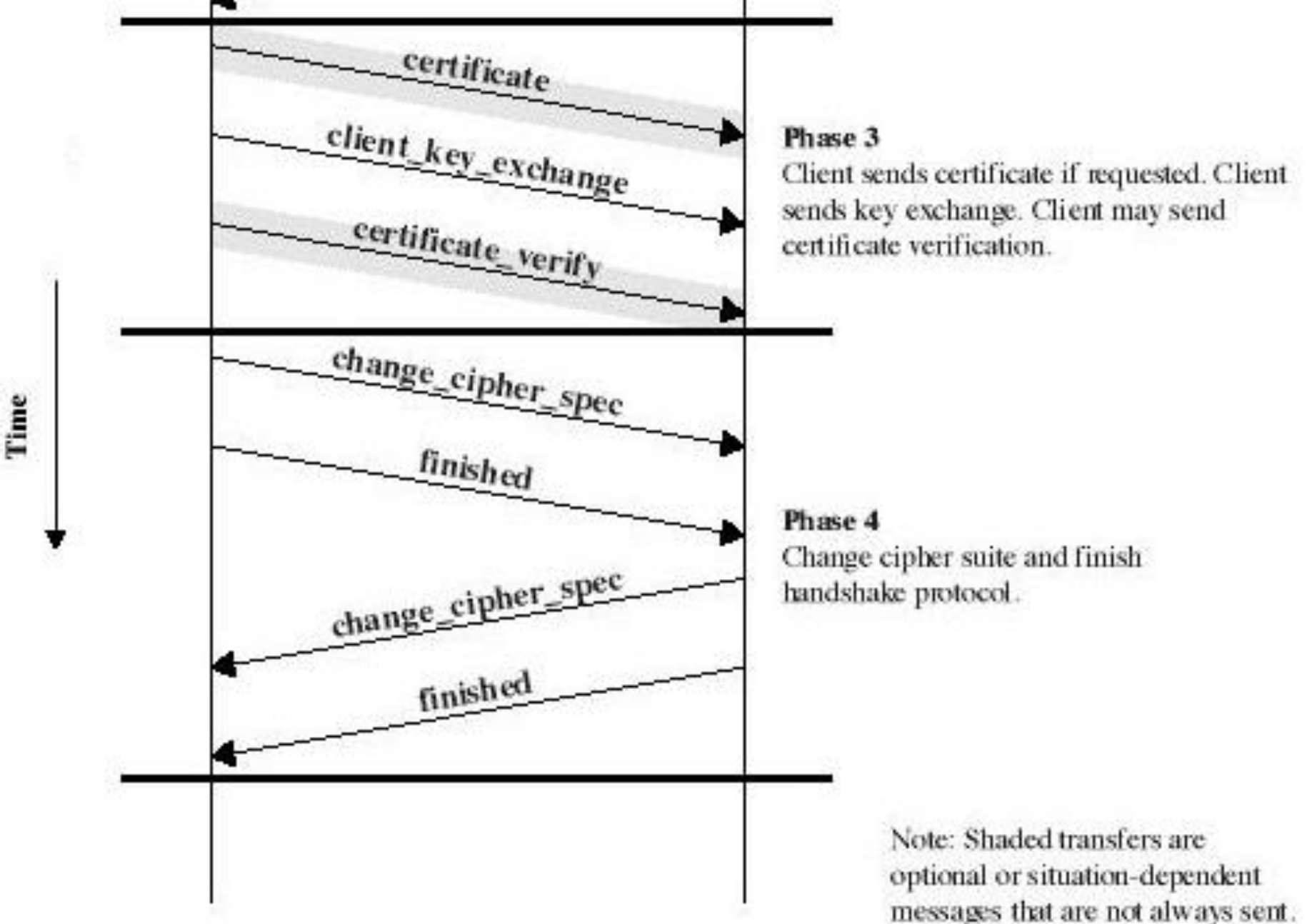
Handshake Payload and Types

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

SSL Handshake Protocol





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 بدون در نظر گرفتن تفاوت‌های جزئی زیر:
 - بهره گیری از HMAC واقعی در محاسبه MAC (استفاده از عملگر XOR)
 - تابع تولید اعداد تصادفی همان HMAC است
 - در TLS کد خطای no-certificate قابل قبول نیست
 - الگوریتم Fortezza از الگوریتم‌های توزیع کلید و رمزگذاری پیغام حذف شده

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
-

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

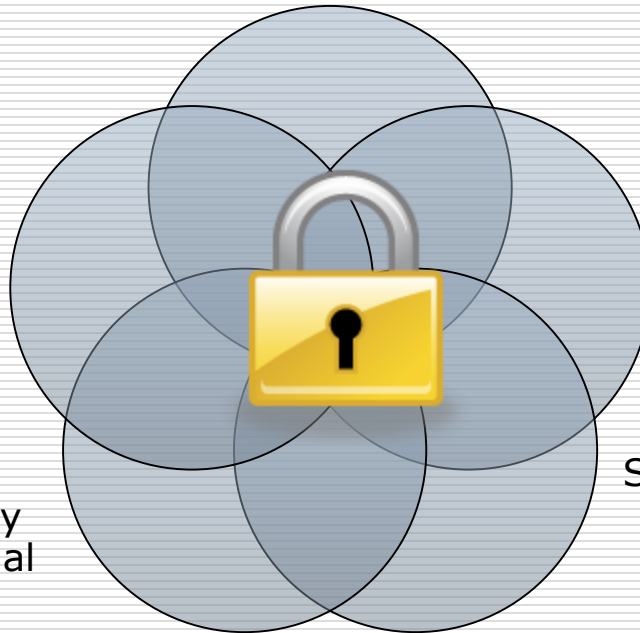
- Has become the method of choice for remote login and X tunneling
- Is rapidly becoming one of the most pervasive applications for encryption technology outside of embedded systems

SSH2 fixes a number of security flaws in the original scheme

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

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

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



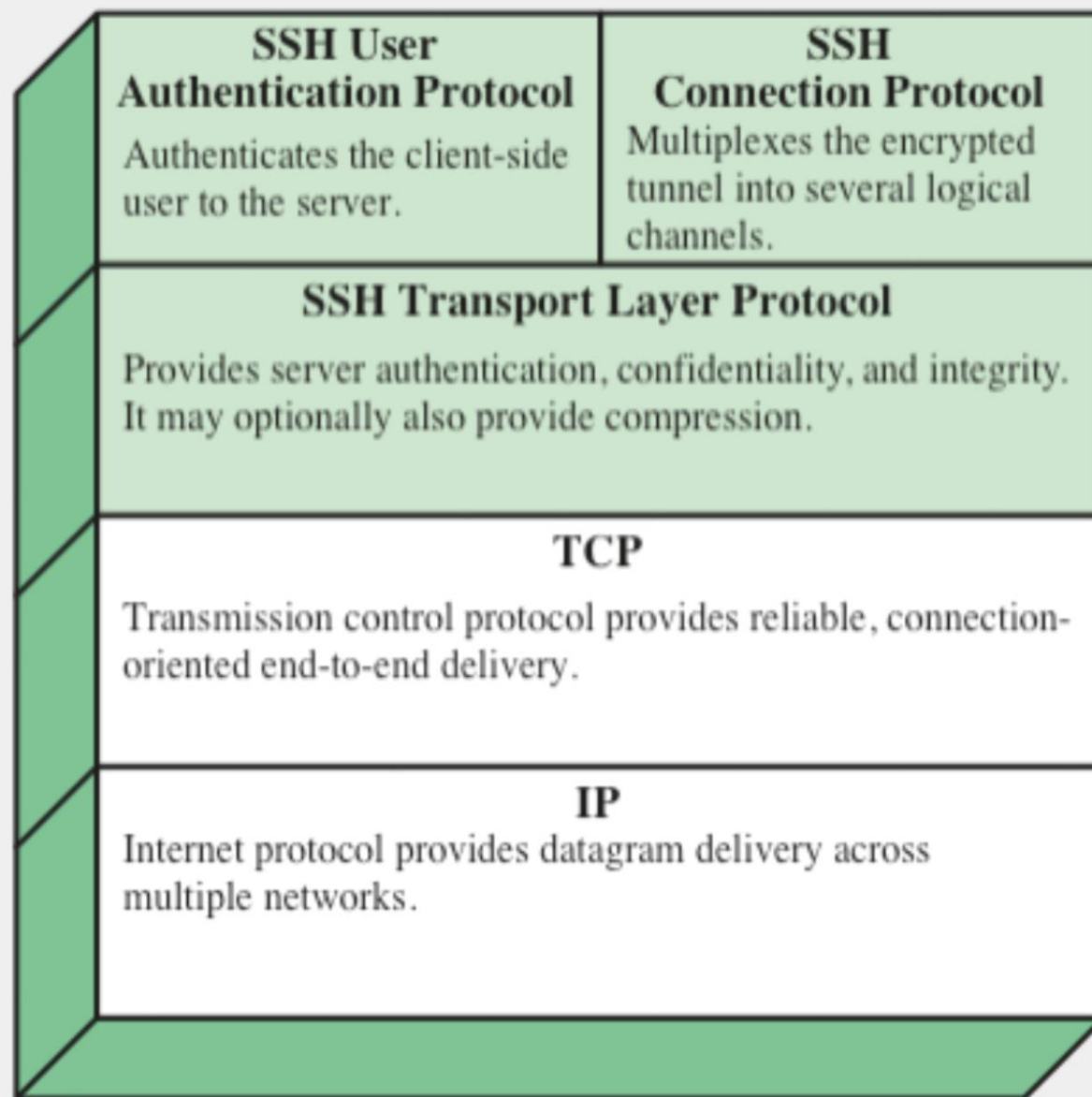


Figure 6.8 SSH Protocol Stack

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
-

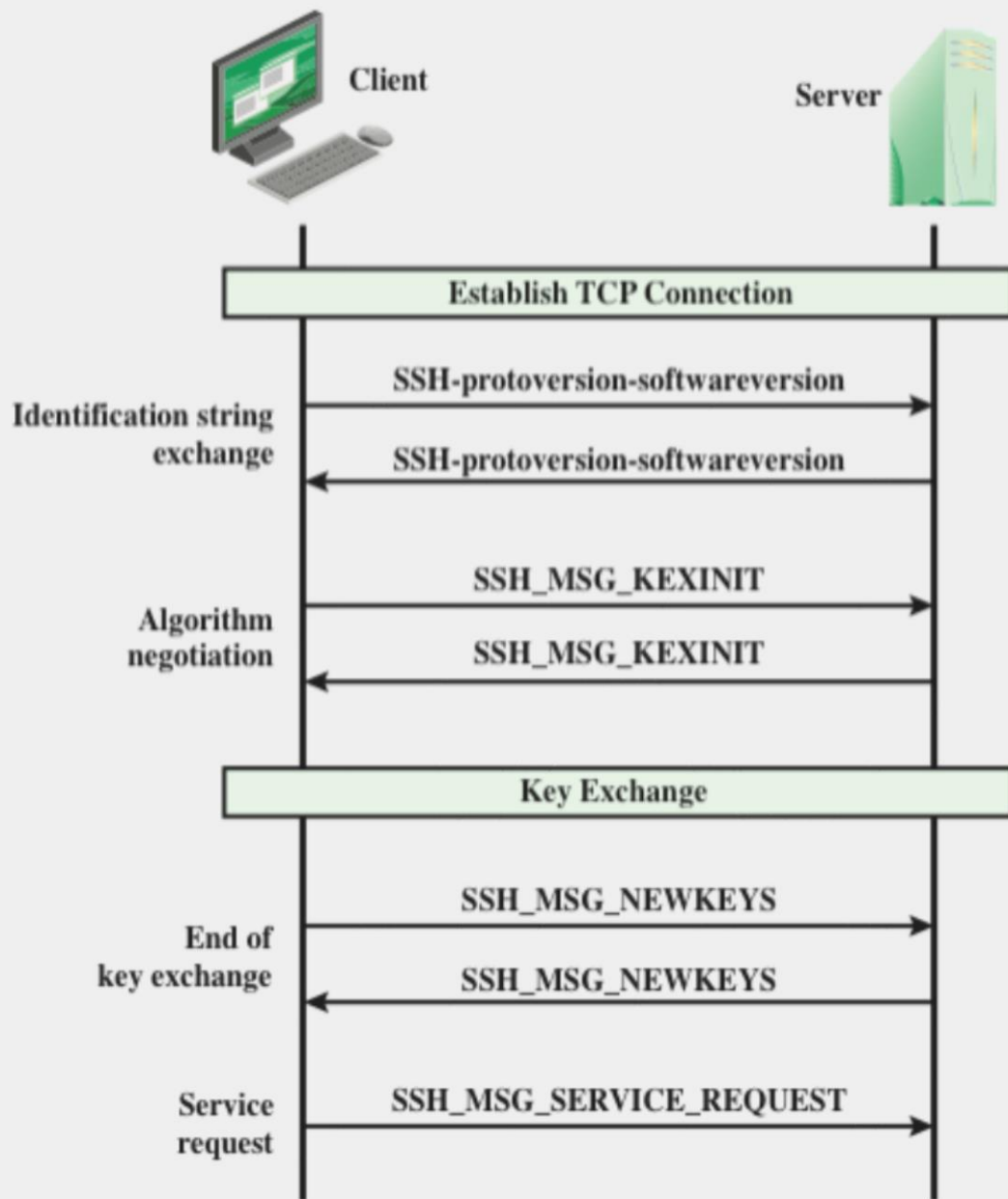
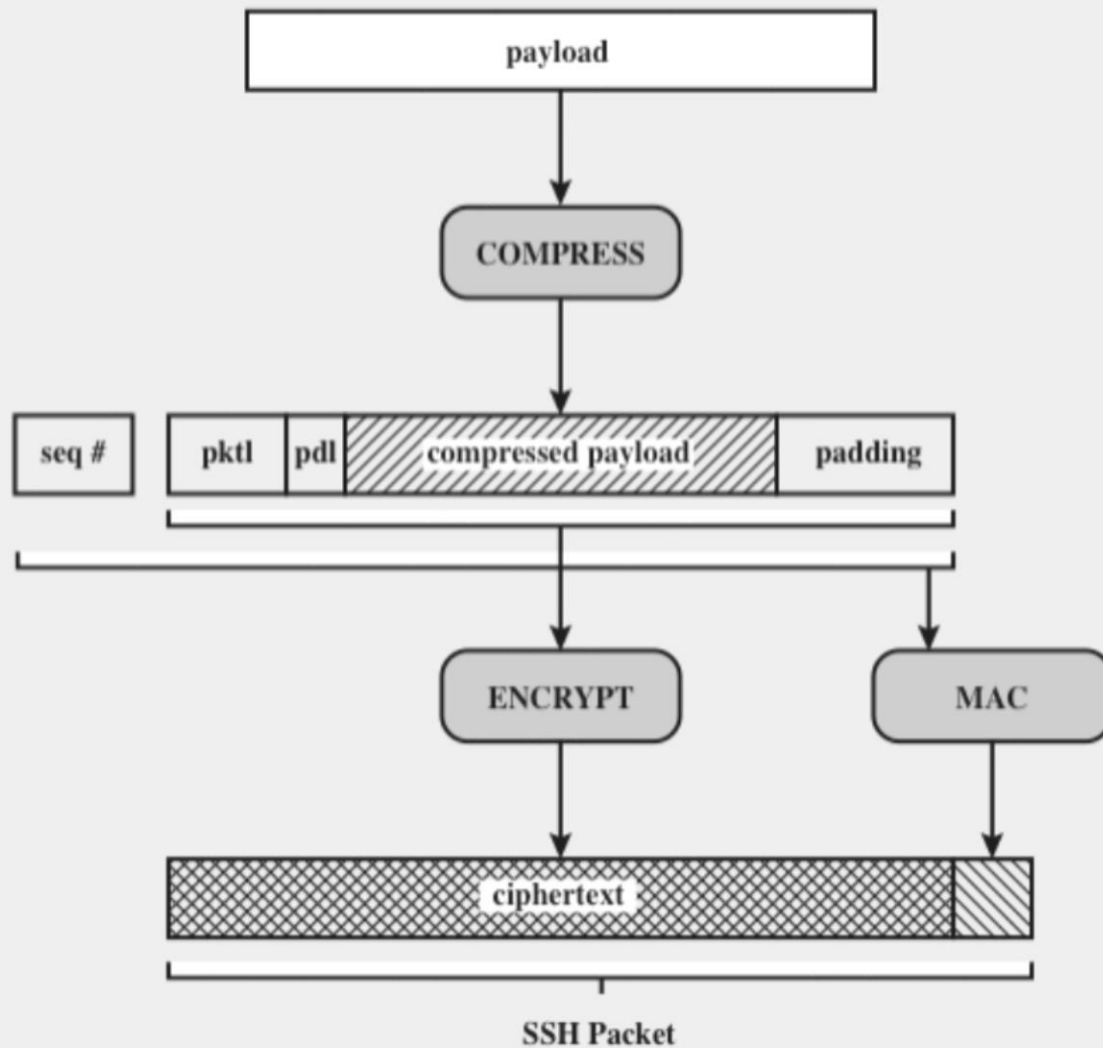


Figure 6.9 SSH Transport Layer Protocol Packet Exchanges



pktl = packet length
pdl = padding length

Figure 6.10 SSH Transport Layer Protocol Packet Formation

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

Table 6.3

SSH

Transport

Layer

Cryptographic

Algorithms

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
-

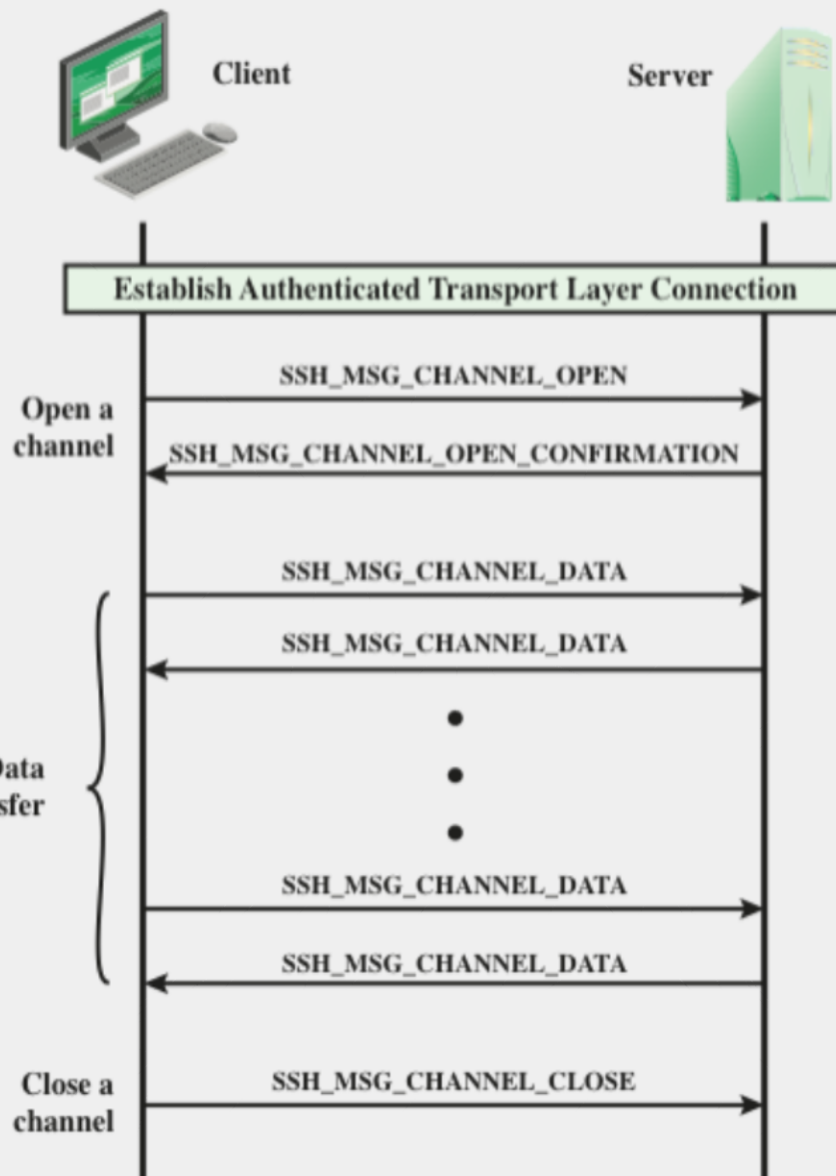


Figure 6.11 Example SSH Connection Protocol Message Exchange

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

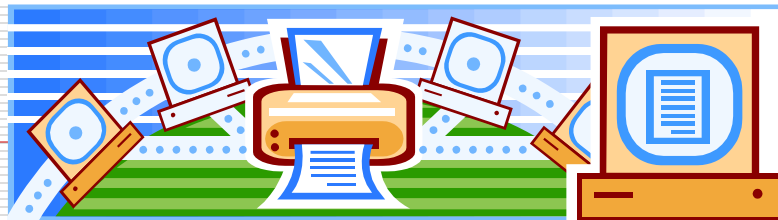
- Remote port forwarding

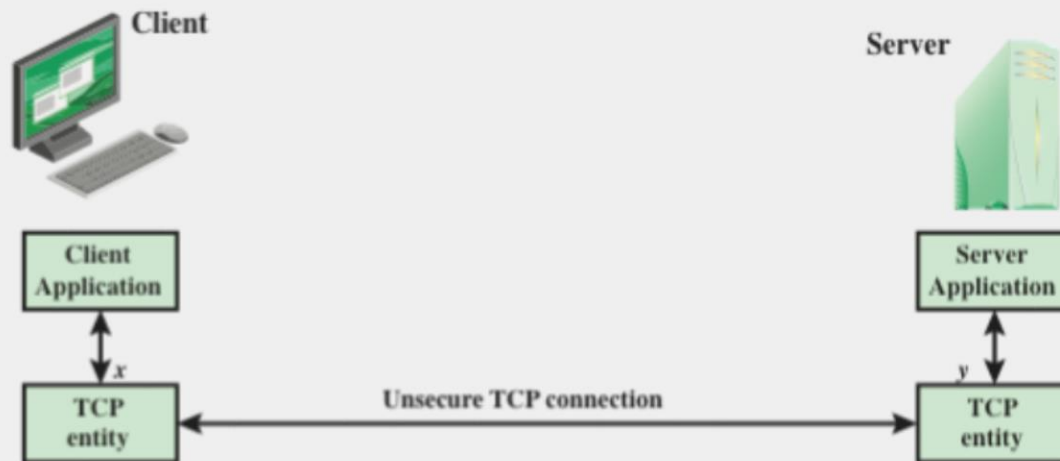
Direct-tcpip

- Local port forwarding

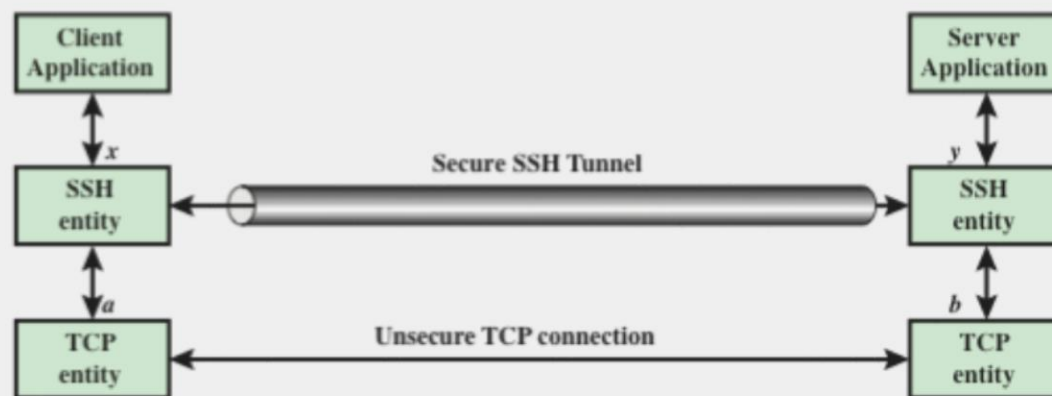
Port Forwarding

- ❑ 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

يا ذا الامن والامان

پایان

