ĐẠI HỌC TÔN ĐỨC THẮNG
TON DUC THANG UNIVERSITY

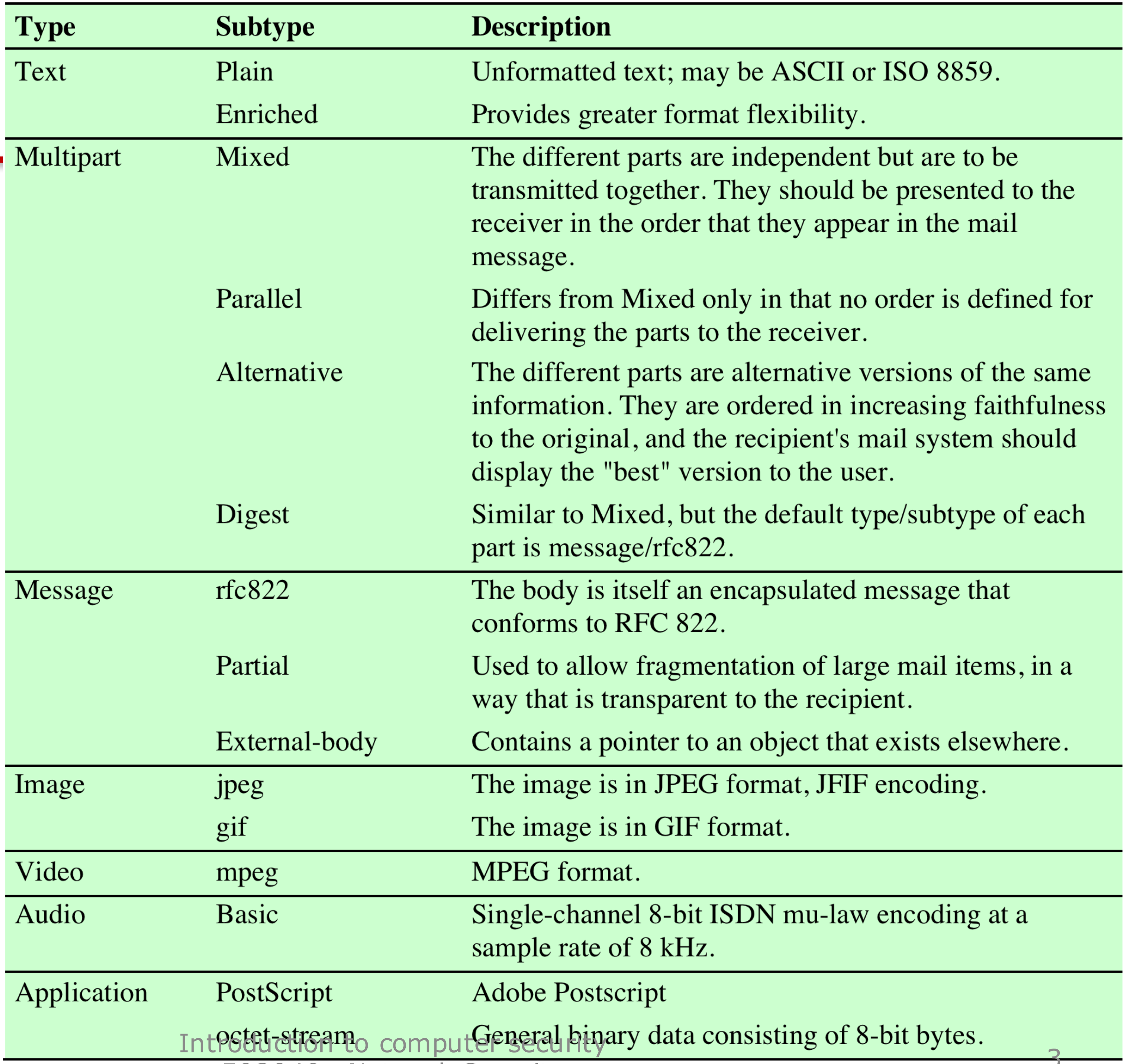
Chapter 11.1
Internet Security Protocols and Standards

Book Reading: Computer Security Principles and Practice (3ed), 2015, p.693-708

S/MIME (Secure/Multipurpose Internet Mail Extensions)

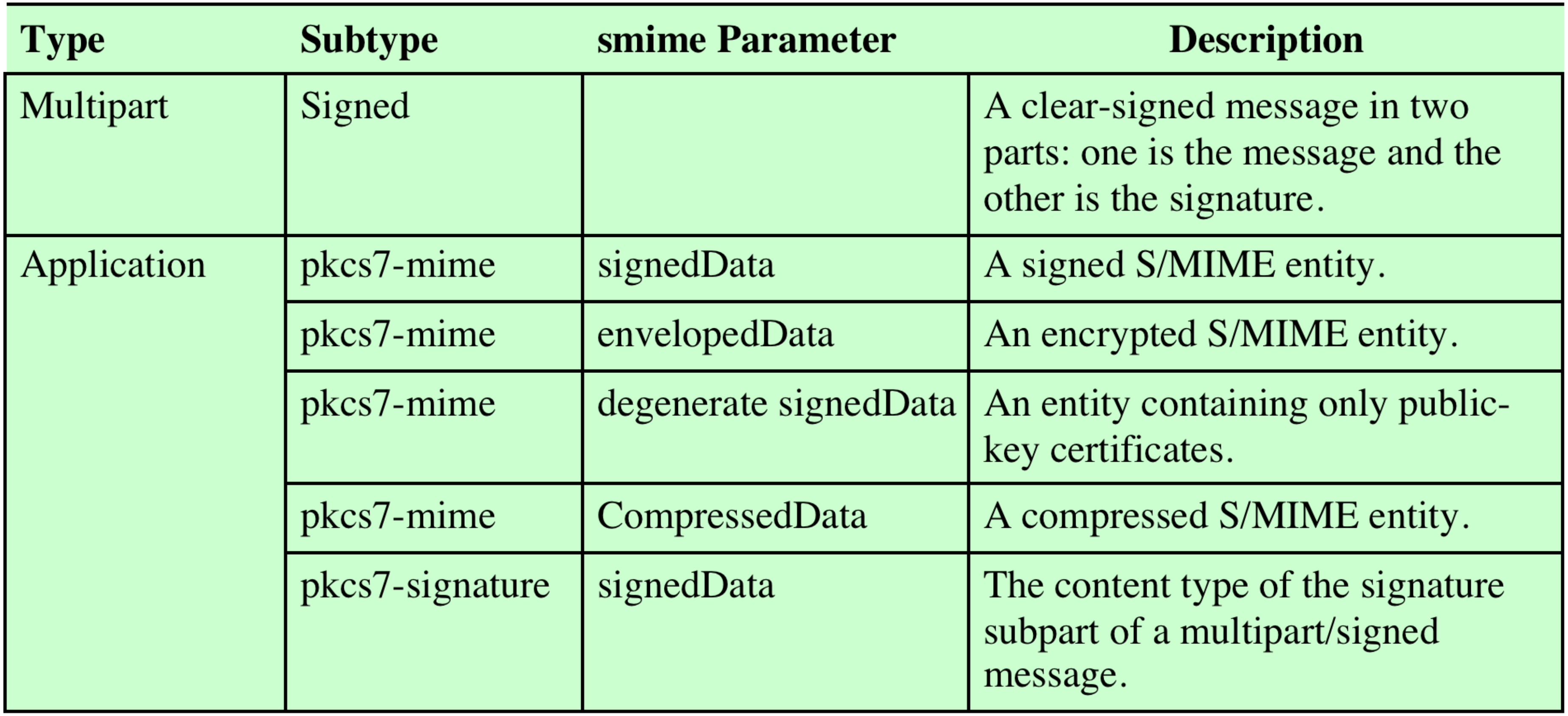
- Security enhancement to MIME email
  - Original Internet RFC822 email was text only
  - MIME provided support for varying content types and multi-part messages
  - With encoding of binary data to textual form
  - S/MIME added security enhancements
- Have S/MIME support (e.g., signed or encryption) in many mail agents
  - MS Outlook, Mozilla, Mac Mail etc

MIME Types



Type	Subtype	Description
Text	Plain	Unformatted text; may be ASCII or ISO 8859.
	Enriched	Provides greater format flexibility.
Multipart	Mixed	The different parts are independent but are to be transmitted together. They should be presented to the receiver in the order that they appear in the mail message.
	Parallel	Differs from Mixed only in that no order is defined for delivering the parts to the receiver.
	Alternative	The different parts are alternative versions of the same information. They are ordered in increasing faithfulness to the original, and the recipient's mail system should display the "best" version to the user.
	Digest	Similar to Mixed, but the default type/subtype of each part is message/rfc822.
Message	rfc822	The body is itself an encapsulated message that conforms to RFC 822.
	Partial	Used to allow fragmentation of large mail items, in a way that is transparent to the recipient.
	External-body	Contains a pointer to an object that exists elsewhere.
Image	jpeg	The image is in JPEG format, JFIF encoding.
	gif	The image is in GIF format.
Video	mpeg	MPEG format.
Audio	Basic	Single-channel 8-bit ISDN mu-law encoding at a sample rate of 8 kHz.
Application	PostScript	Adobe Postscript
	octet-stream	General binary data consisting of 8-bit bytes.

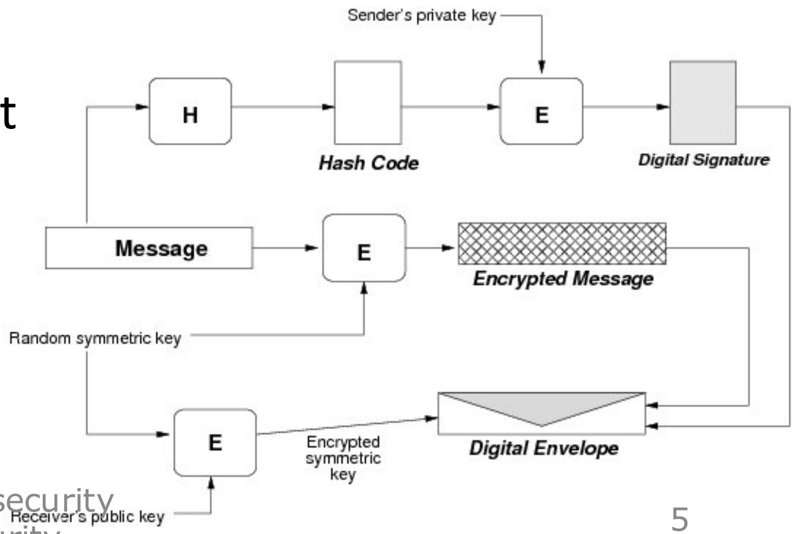
S/MIME Types



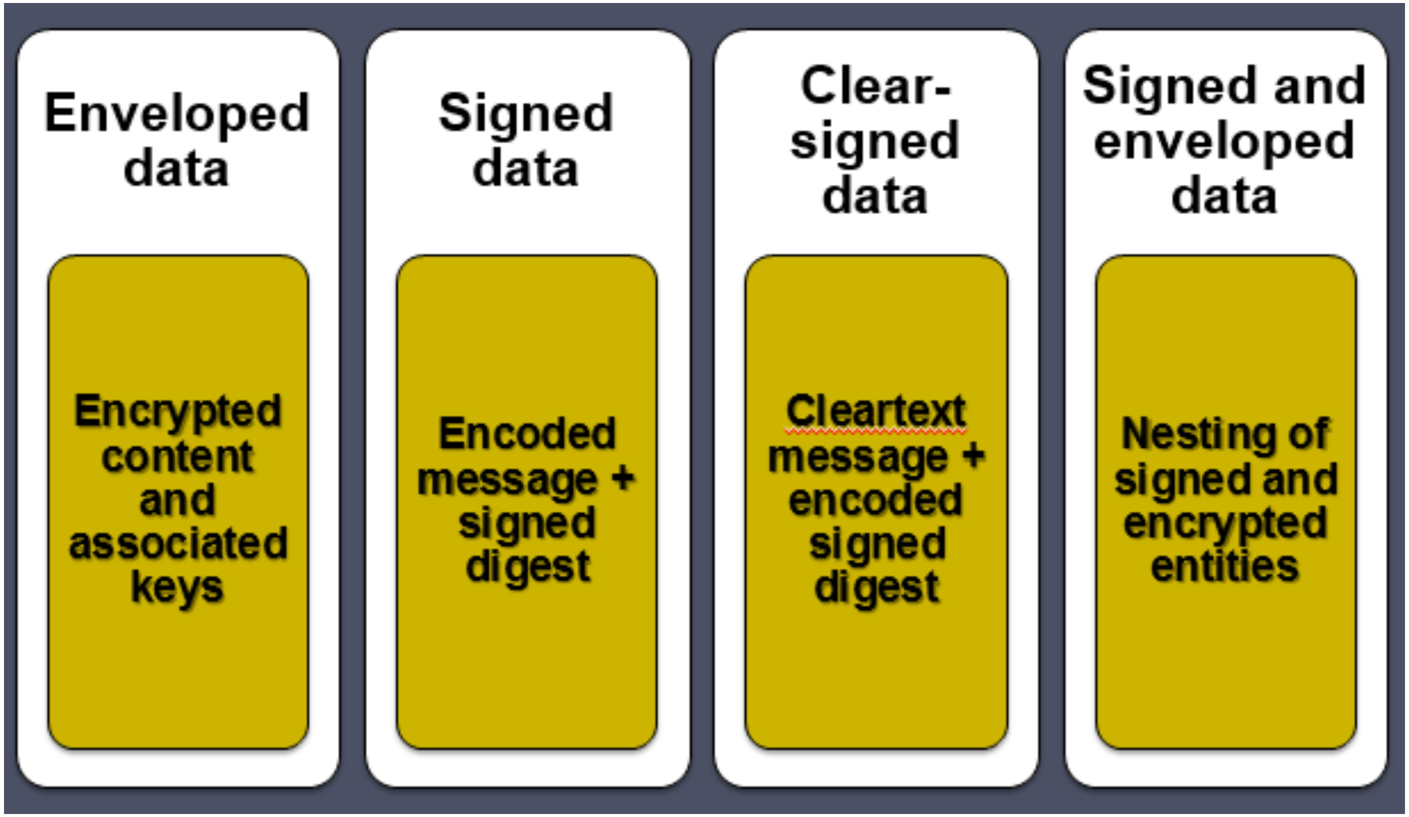
Type	Subtype	smime Parameter	Description
Multipart	Signed		A clear-signed message in two parts: one is the message and the other is the signature.
Application	pkcs7-mime	signedData	A signed S/MIME entity.
	pkcs7-mime	envelopedData	An encrypted S/MIME entity.
	pkcs7-mime	degenerate signedData	An entity containing only public-key certificates.
	pkcs7-mime	CompressedData	A compressed S/MIME entity.
	pkcs7-signature	signedData	The content type of the signature subpart of a multipart/signed message.

S/MIME Functions

- Enveloped data
  - encrypted content and associated keys
- Signed data
  - digital signature (hash code of msg encrypted with sender’s private key)
  - encrypted msg (random sym key later encrypted)
  - can be view by recipient with S/MIME
- Clear-signed data
  - cleartext message + encoded signed digest
- Signed & enveloped data
  - nesting of signed & encrypted entities

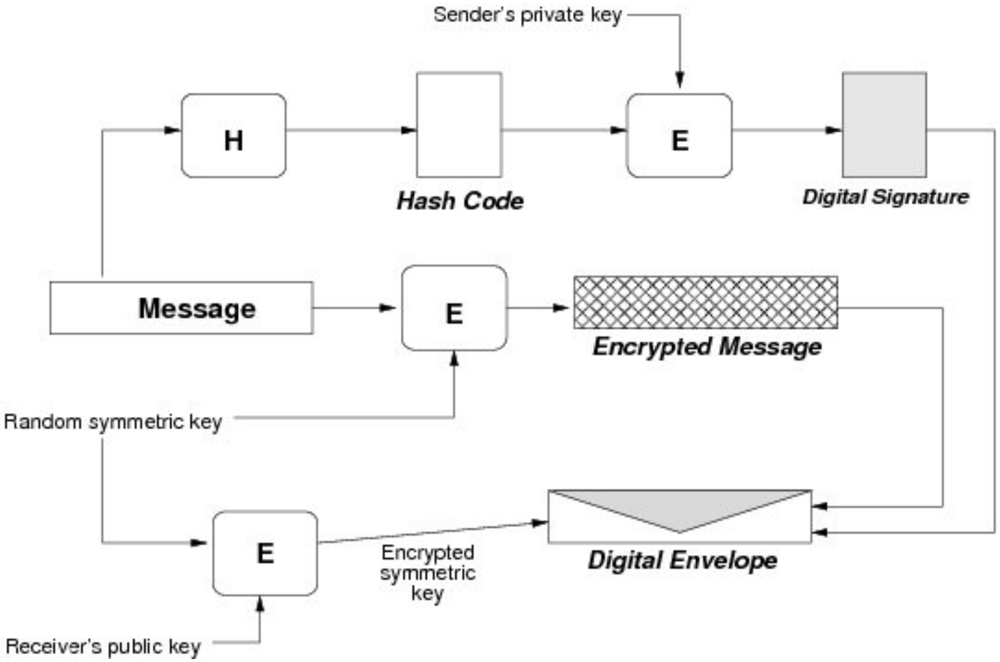


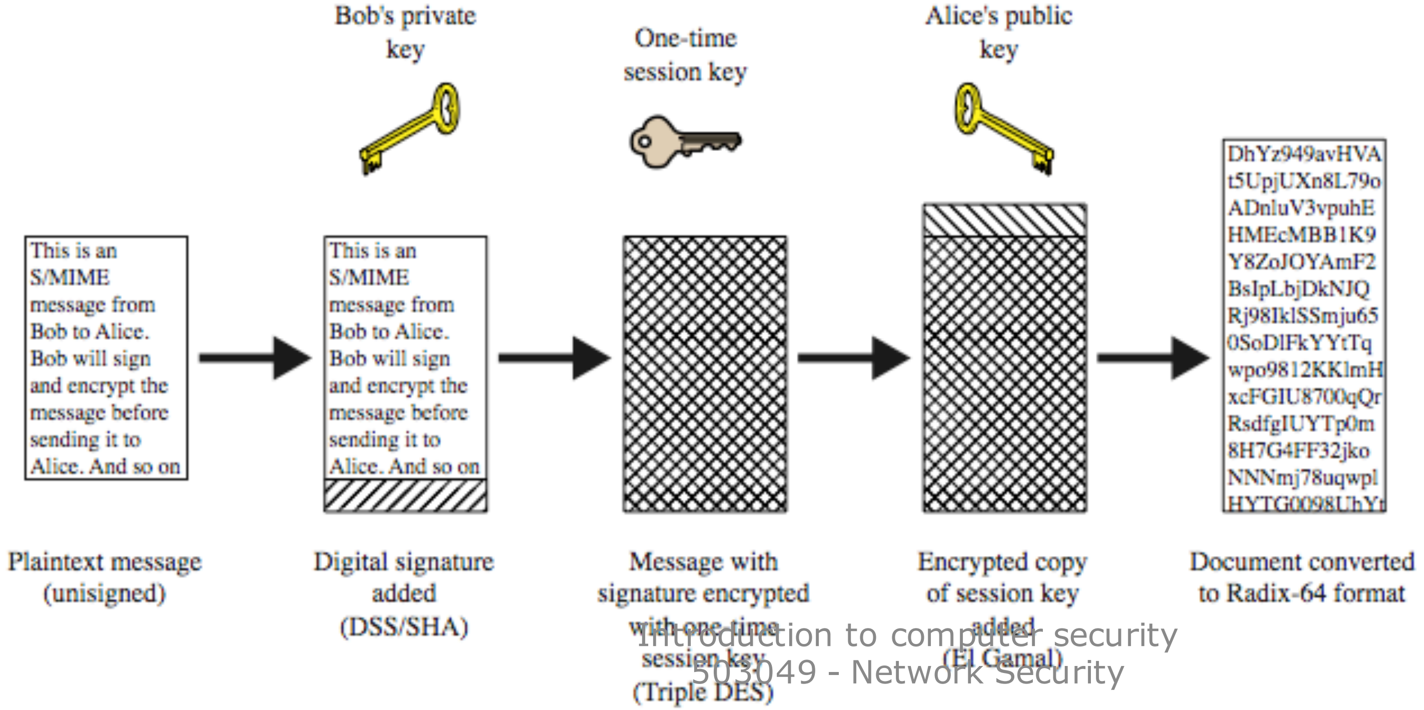
S/MIME Functions (Summary)



Enveloped data	Signed data	Clear-signed data	Signed and enveloped data
Encrypted content and associated keys	Encoded message + signed digest	Cleartext message + encoded signed digest	Nesting of signed and encrypted entities

S/MIME Process




Bob's private key
One-time session key
Alice's public key
Plaintext message (unsigned)
Digital signature added (DSS/SHA)
Message with signature encrypted with one-time session key (Triple DES)
Encrypted copy of session key added (El Gamal)
Document converted to Radix-64 format

This is an S/MIME message from Bob to Alice. Bob will sign and encrypt the message before sending it to Alice. And so on

This is an S/MIME message from Bob to Alice. Bob will sign and encrypt the message before sending it to Alice. And so on

```
DhYz949avHVA
t5UpjUXn8L79o
ADnluV3vpuhE
HMEcMBB1K9
Y8ZoJOYAmF2
BsIpLbjDkNJQ
Rj98IkISSmju65
0SoDIFkYYtTq
wpo9812KKlmH
xcFGIU8700qQr
RsdfgIUYTp0m
8H7G4FF32jko
NNNmj78uqwpl
HYTG0098UhYt
```

S/MIME Cryptographic Algorithms

- Digital signatures: DSS & RSA
- Hash functions: SHA-1 & MD5
- Session key encryption: ElGamal & RSA
- Message encryption: AES, 3DES, etc
- MAC: HMAC with SHA-1
- Must map binary values to printable ASCII
  - use radix-64 or base64 mapping

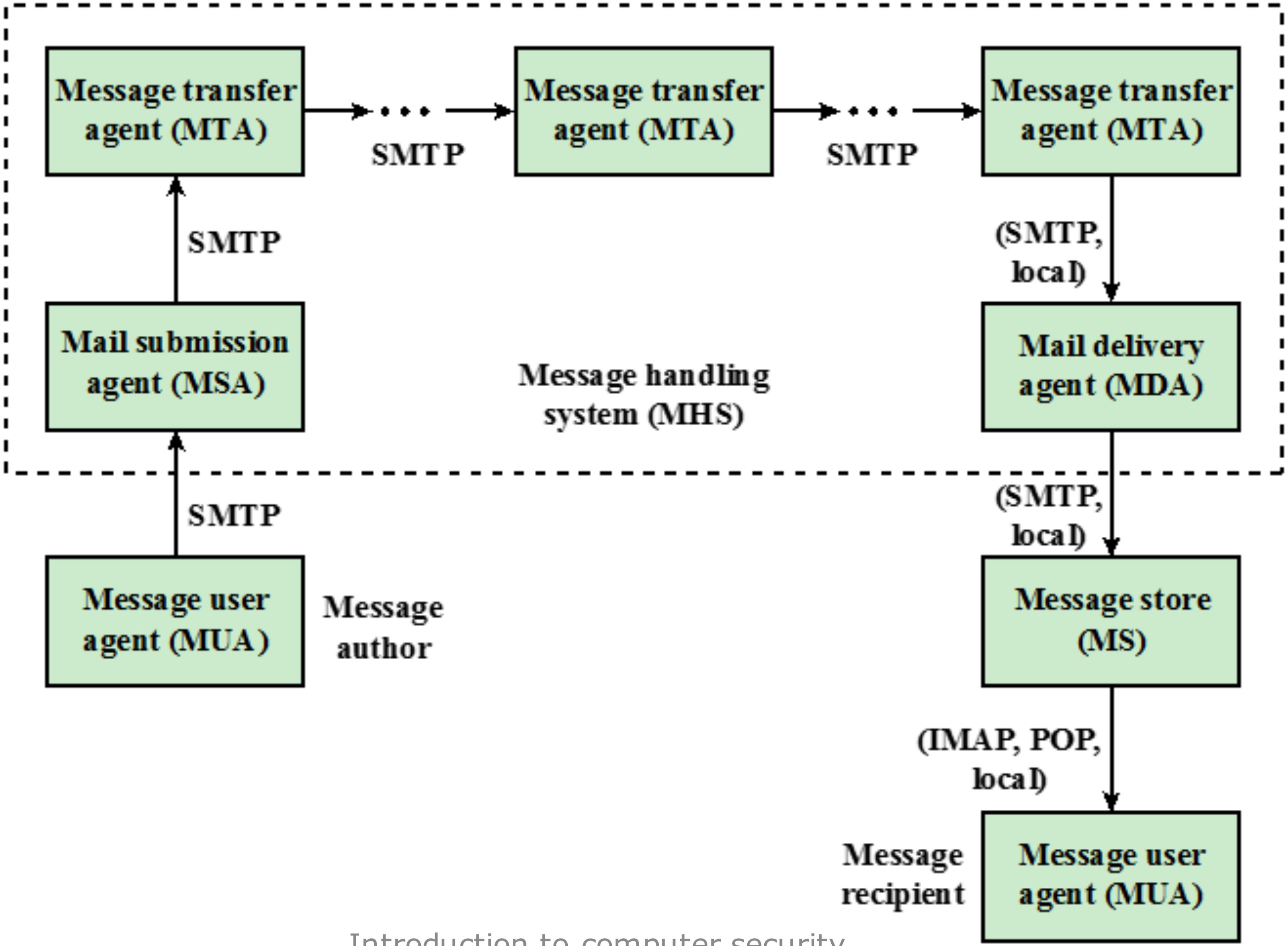
S/MIME Public Key Certificates

- S/MIME has effective encryption and signature services
- But also need to manage public-keys
- S/MIME uses X.509 v3 certificates
- Each client has a list of trusted CA’s certs
- And own public/private key pairs & certs
- Certificates must be signed by trusted CA’s

DomainKeys Identified Mail (DKIM)

- Specification of cryptographically signing e-mail messages permitting a signing domain to claim responsibility for a message in the mail stream
- Proposed Internet Standard: *DomainKeys Identified Mail (DKIM) Signatures*
- Widely adopted by a range of e-mail providers

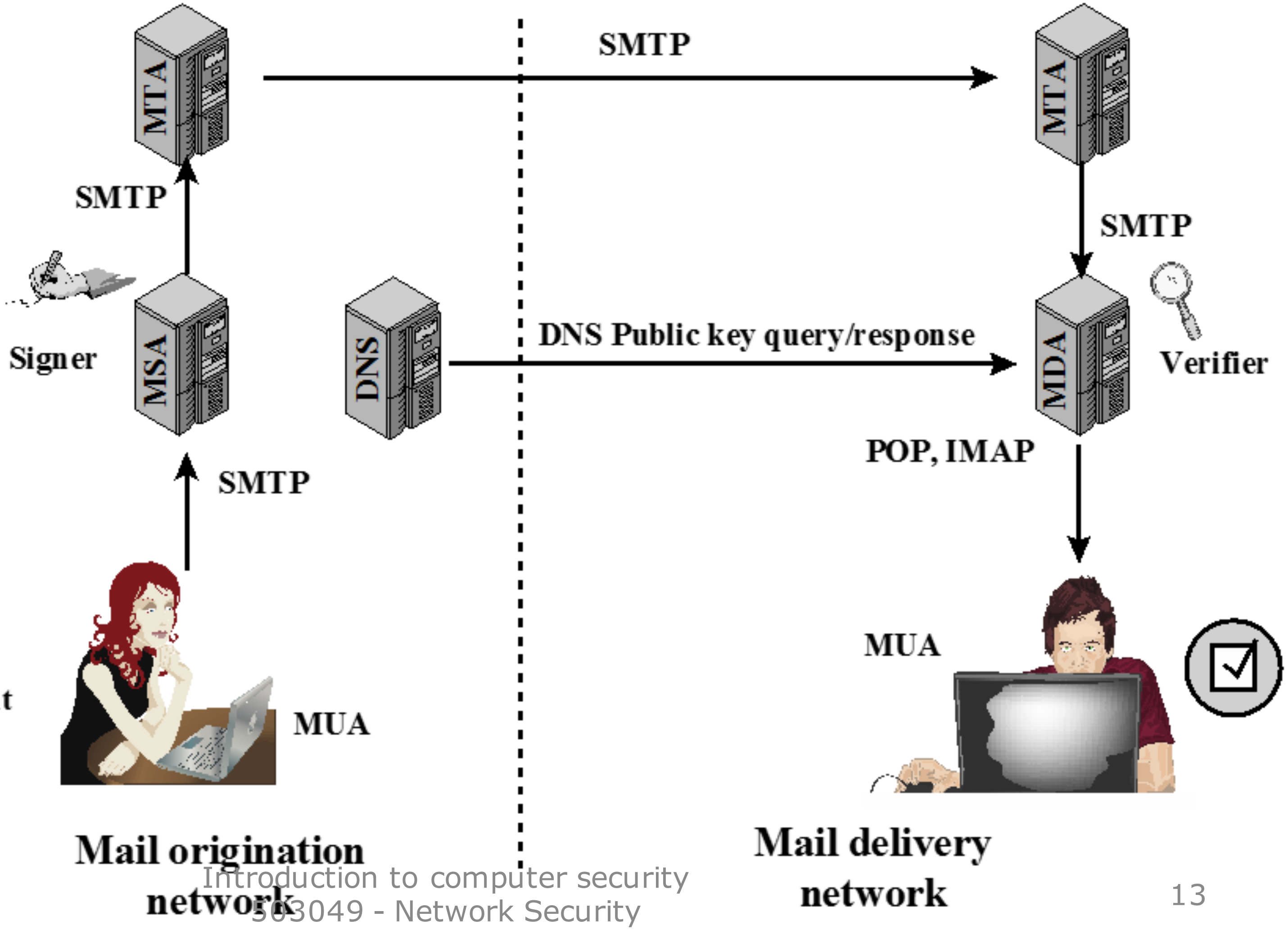
Internet E-Mail Architecture


Message transfer agent (MTA)
SMTP
Message transfer agent (MTA)
SMTP
Message transfer agent (MTA)
SMTP
(SMTP, local)
Mail submission agent (MSA)
Message handling system (MHS)
Mail delivery agent (MDA)
SMTP
(SMTP, local)
Message user agent (MUA)
Message author
Message store (MS)
(IMAP, POP, local)
Message recipient
Message user agent (MUA)

DKIM Purpose

- Email authentication between hosts
- Transparent to the user
- Each email is signed by the private key of the administrative domain
- To authenticate that the message comes from the claimed administrative domain
- Mail delivery agent (MDA) does the verification

Domain Keys Identified Mail (DKIM) Deployment



DNS = domain name system
MDA = mail delivery agent
MSA = mail submission agent
MTA = message transfer agent
MUA = message user agent

DKIM vs S/MIME

- S/MIME depends on both sender and receiver users using S/MIME (many users however don’t user it)
- S/MIME only signs the contents; email header may be compromised
- DKIM is not implemented in client programs (MUAs) – transparent to the users
- DKIM applies to all emails
- DKIM allows good senders to prove they sent a particular message

Internet Security Protocols and Standards

- Secure Sockets Layer (SSL) and the follow-up Transport Layer Security (TLS)
  - SSL: a general-purpose set of protocols; relies on TCP
  - Full generality as part of the a protocol suite (transparent to apps)
  - Embedded in a specific app (e.g., IE)
- IPv4 and IPv6 (IP level security)
- HTTPS

Secure Sockets Layer (SSL)

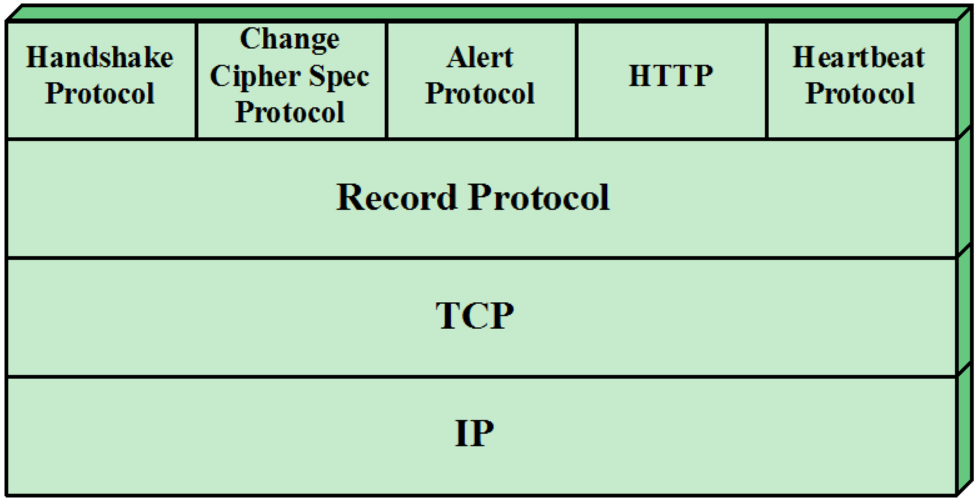
- Transport layer security service
  - originally developed by Netscape
  - version 3 designed with public input
- Subsequently became Internet standard Transport Layer Security (TLS)
- Use TCP to provide a reliable end-to-end service
- May be provided in underlying protocol suite transparent to apps)
- Or embedded in specific packages (WWW browsers)

SSL Protocol Stack

SSL Rec Protocol provides Basic sec services to the above Protocols (eg, HTTP):

- HTTP provides xfer svc for Web client/server interactions

- Other three protocols are used in management of SSL

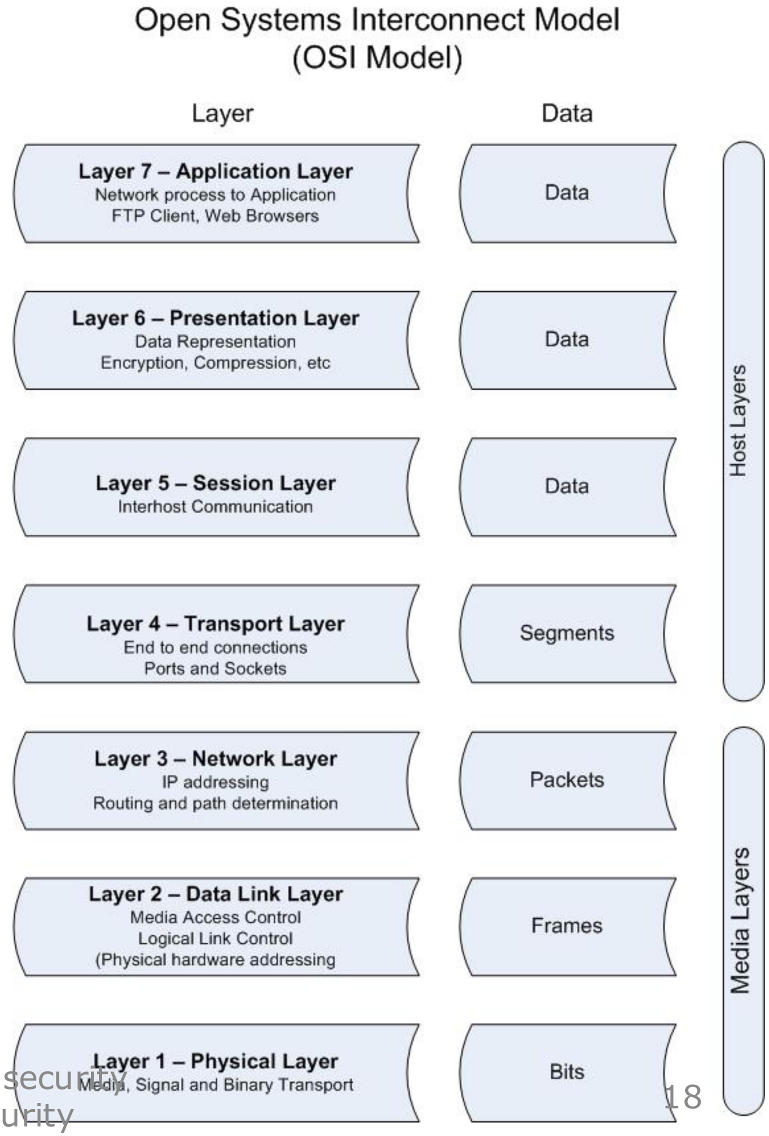


Handshake Protocol
Change Cipher Spec Protocol
Alert Protocol
HTTP
Heartbeat Protocol
Record Protocol
TCP
IP

Two Important SSL Concepts

- SSL session: an association between a client and a server created by the Handshake protocol

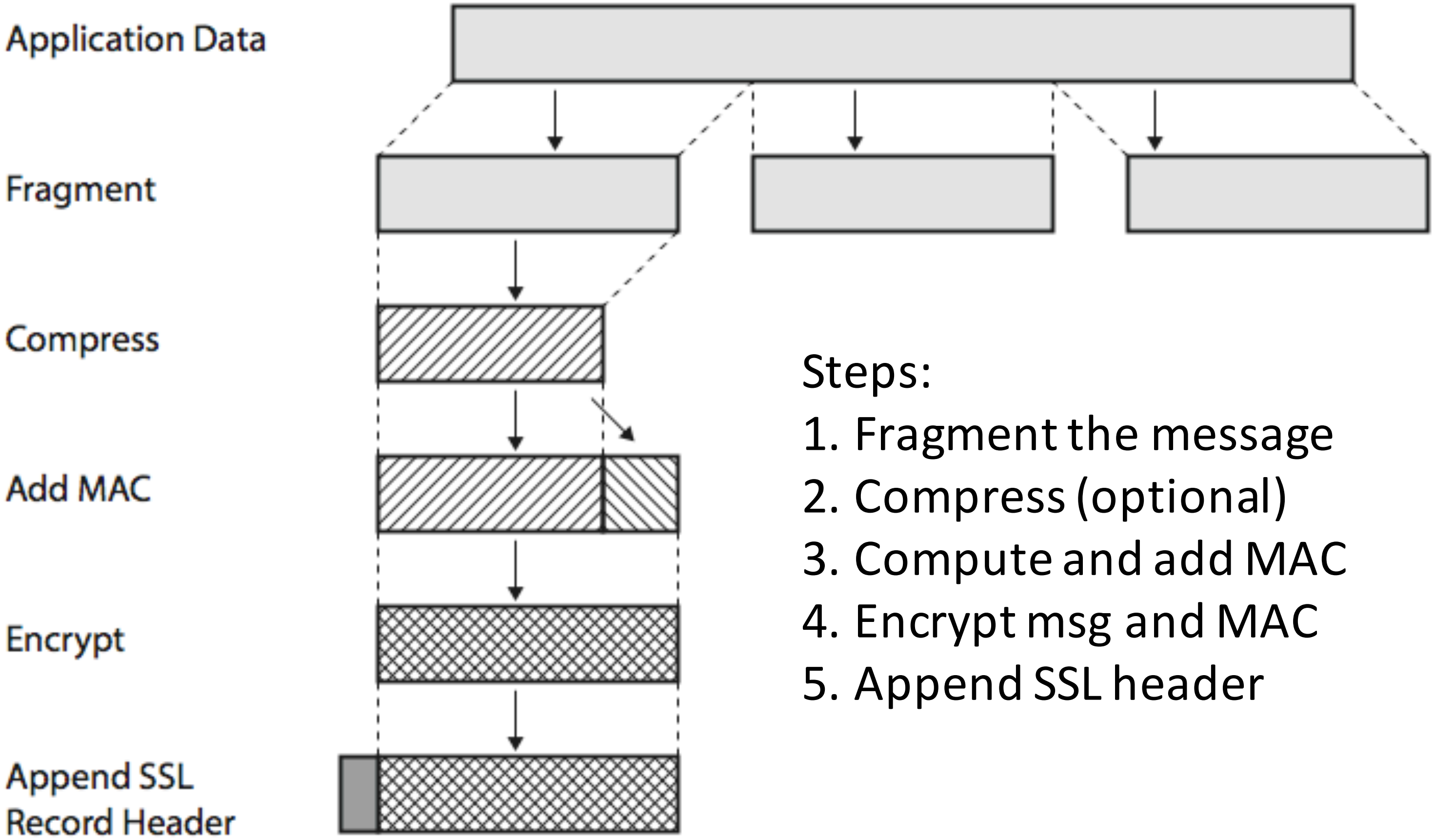
- SSL connection: a transport layer, peer-to-peer, short-live connection
  - Every connection is associated with one session



SSL Record Protocol Services

- **Confidentiality**
  - – The Handshake Protocol defines a shared secret key for symmetric encryption of SSL payloads
  - – AES, IDEA, RC2-40, DES-40, DES, 3DES, Fortezza, RC4-40, RC4-128
  - – Message is compressed before encryption
- **Message integrity**
  - – the Handshake Protocol also defines a shared secret key to form a msg authentication code (MAC)

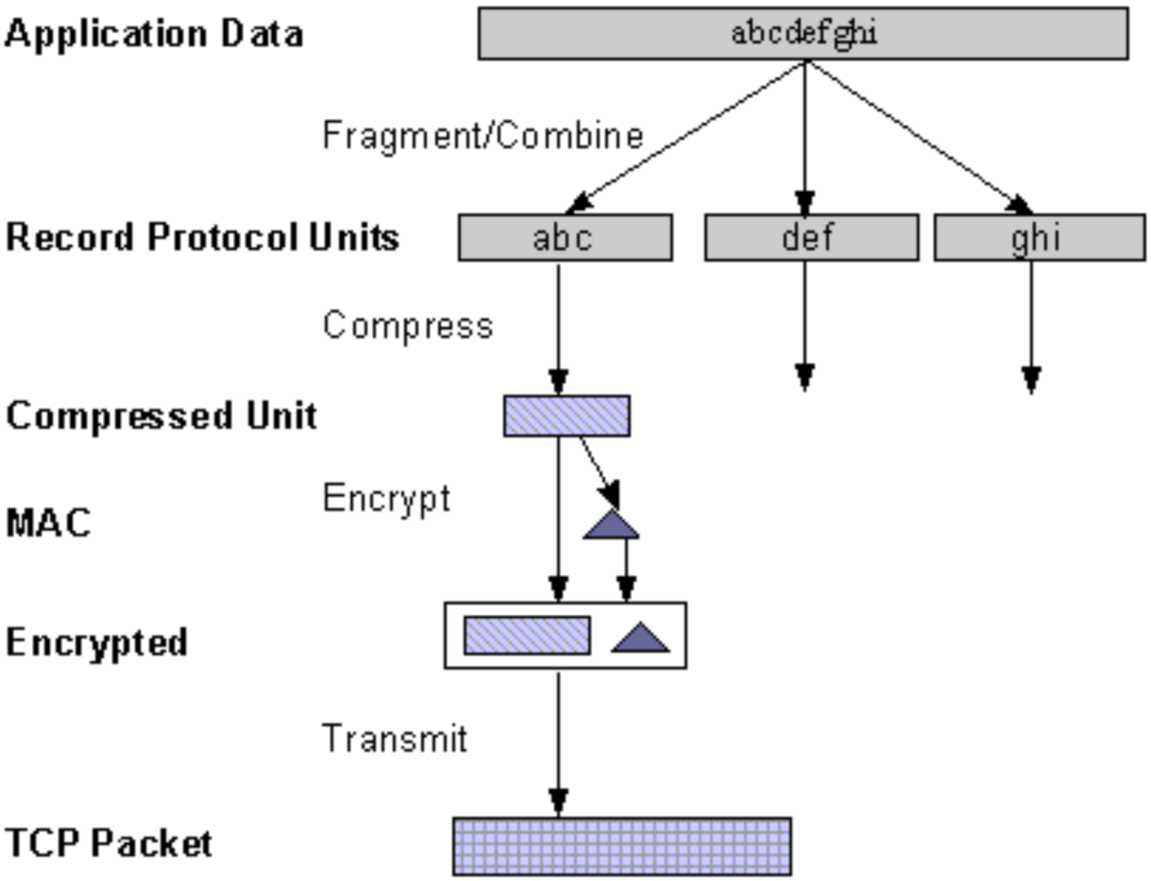
SSL Record Protocol Operation



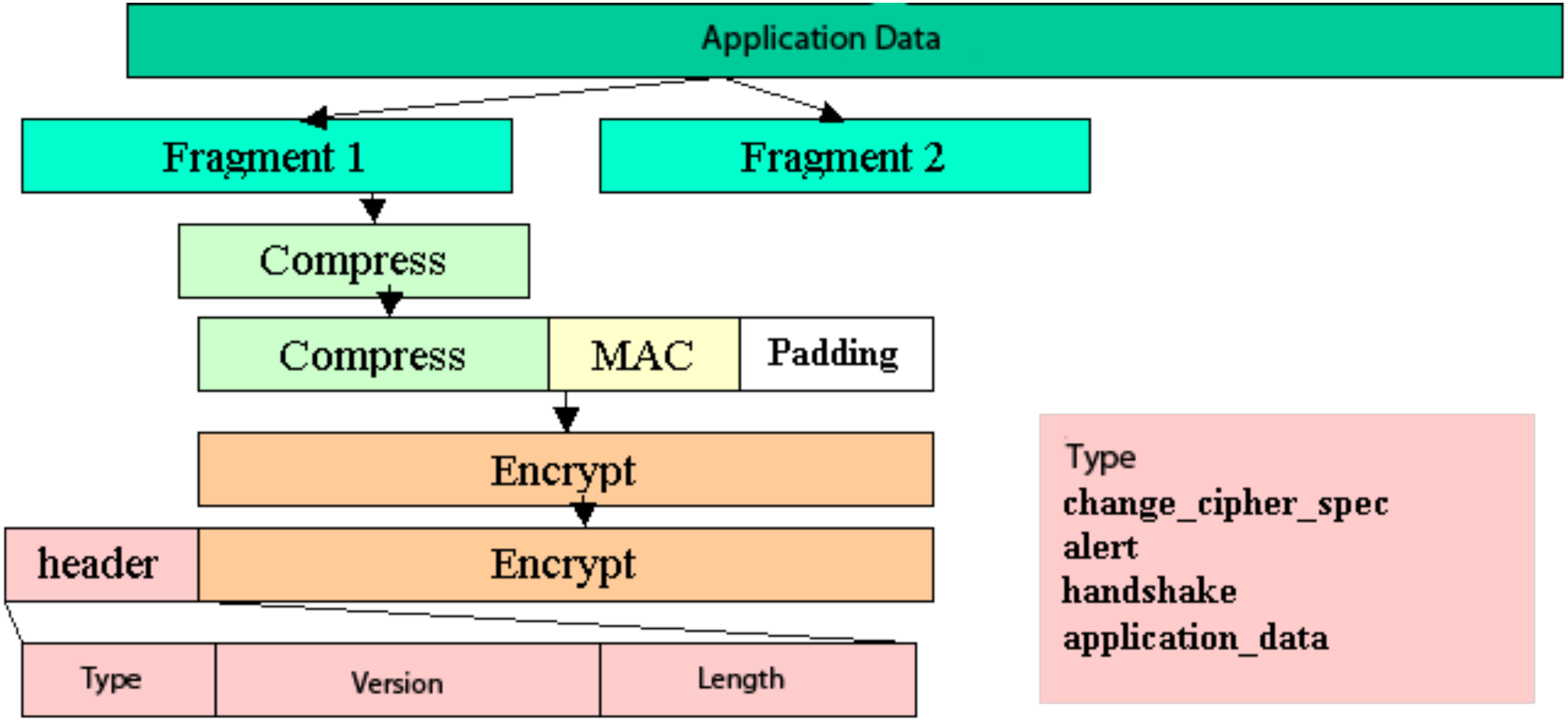
Steps:

1. Fragment the message
2. Compress (optional)
3. Compute and add MAC
4. Encrypt msg and MAC
5. Append SSL header

SSL Record Protocol Operation



SSL Record Protocol Operation



Type
change_cipher_spec
alert
handshake
application_data

SSL Change Cipher Spec Protocol

- One of 3 SSL specific protocols which use the SSL Record protocol
- A single message (a single byte with value 1)
- Causes pending state to become current (last stage of a handshake)
- Hence updating the cipher suite in use (collection of cryptographic algorithms supported by both parties)

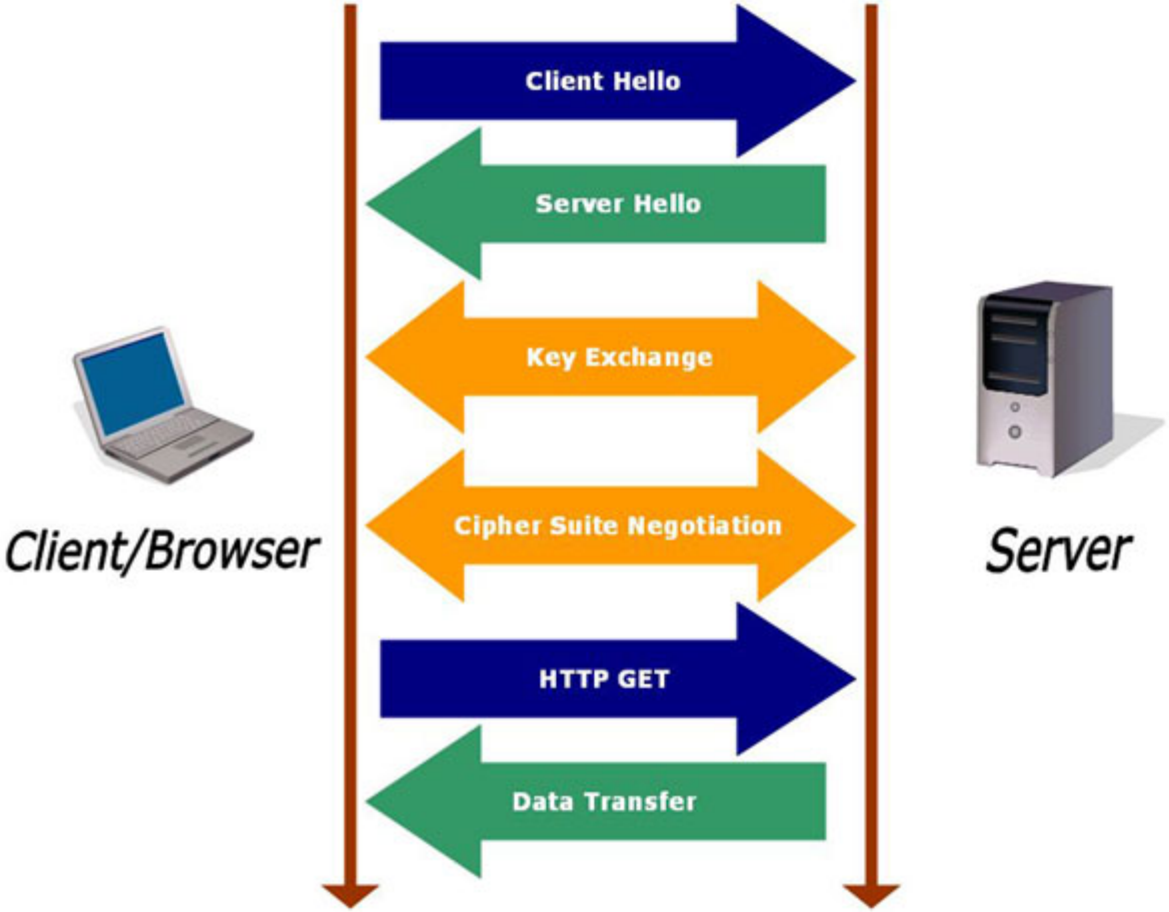
SSL Alert Protocol

- Conveys SSL-related alerts to peer entity
- Severity: warning or fatal
- Specific alert
  - fatal: unexpected message, bad record mac, decompression failure, handshake failure, illegal parameter
  - warning: close notify, no certificate, bad certificate, unsupported certificate, certificate revoked, certificate expired, certificate unknown
- Compressed & encrypted like all SSL data

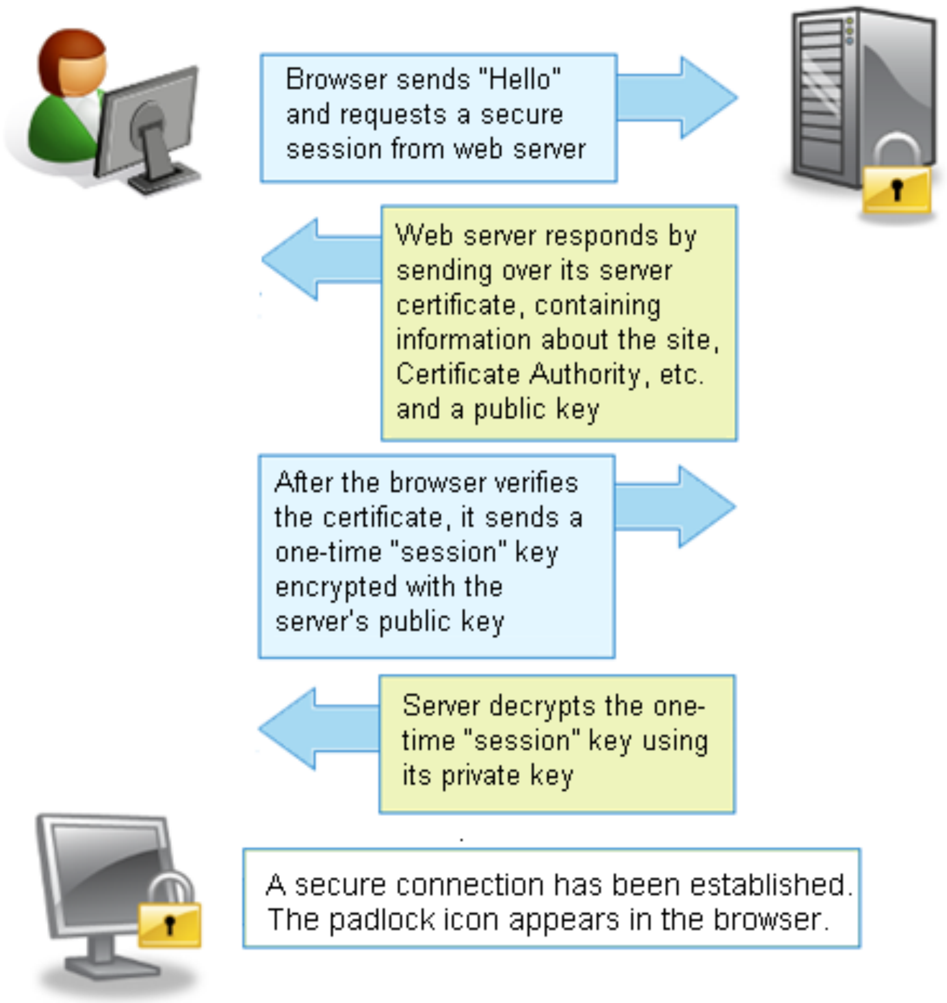
SSL Handshake Protocol

- Allows server & client to:
  - authenticate each other
  - to negotiate encryption & MAC algorithms
  - to negotiate cryptographic keys to be used
- Comprises a series of messages in phases
  1. Establish Security Capabilities
  2. Server Authentication and Key Exchange
  3. Client Authentication and Key Exchange
  4. Finish

SSL Handshake Protocol



SSL Handshake Protocol



Browser sends "Hello" and requests a secure session from web server

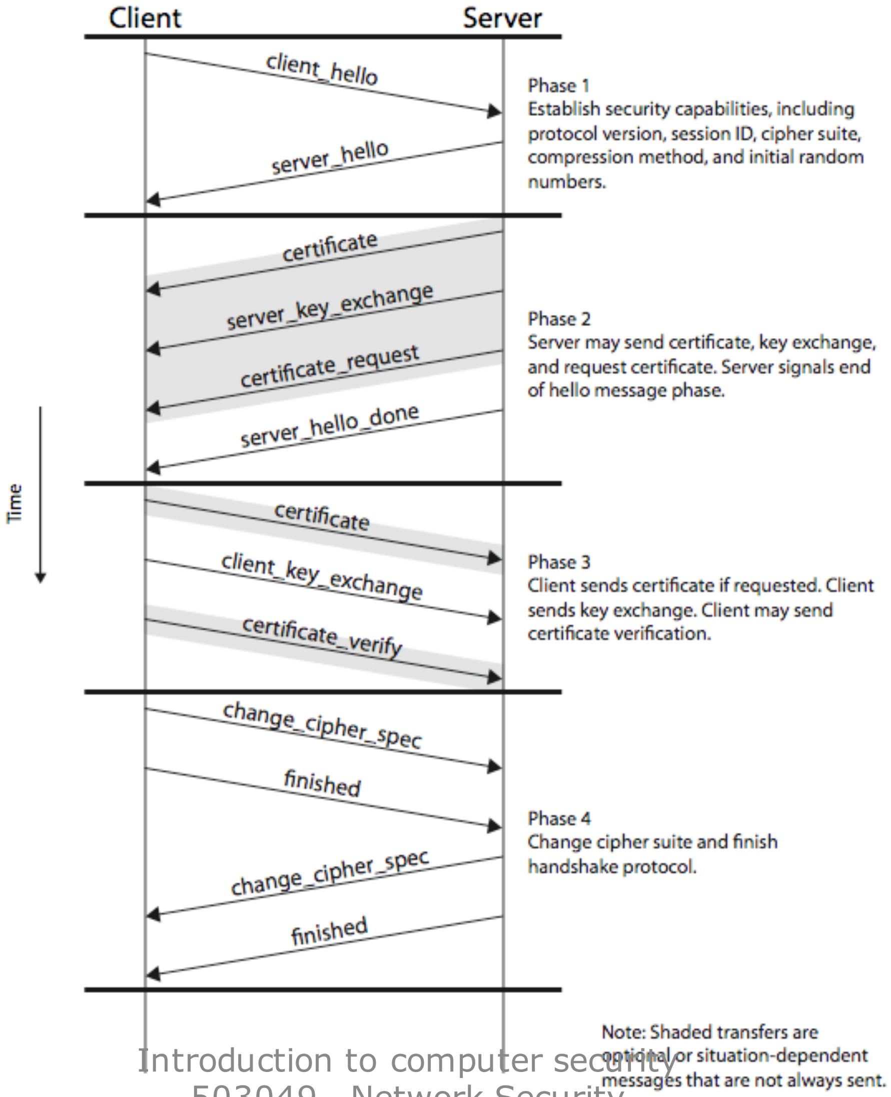
Web server responds by sending over its server certificate, containing information about the site, Certificate Authority, etc. and a public key

After the browser verifies the certificate, it sends a one-time "session" key encrypted with the server's public key

Server decrypts the one-time "session" key using its private key

A secure connection has been established. The padlock icon appears in the browser.

SSL Handshake Protocol



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.

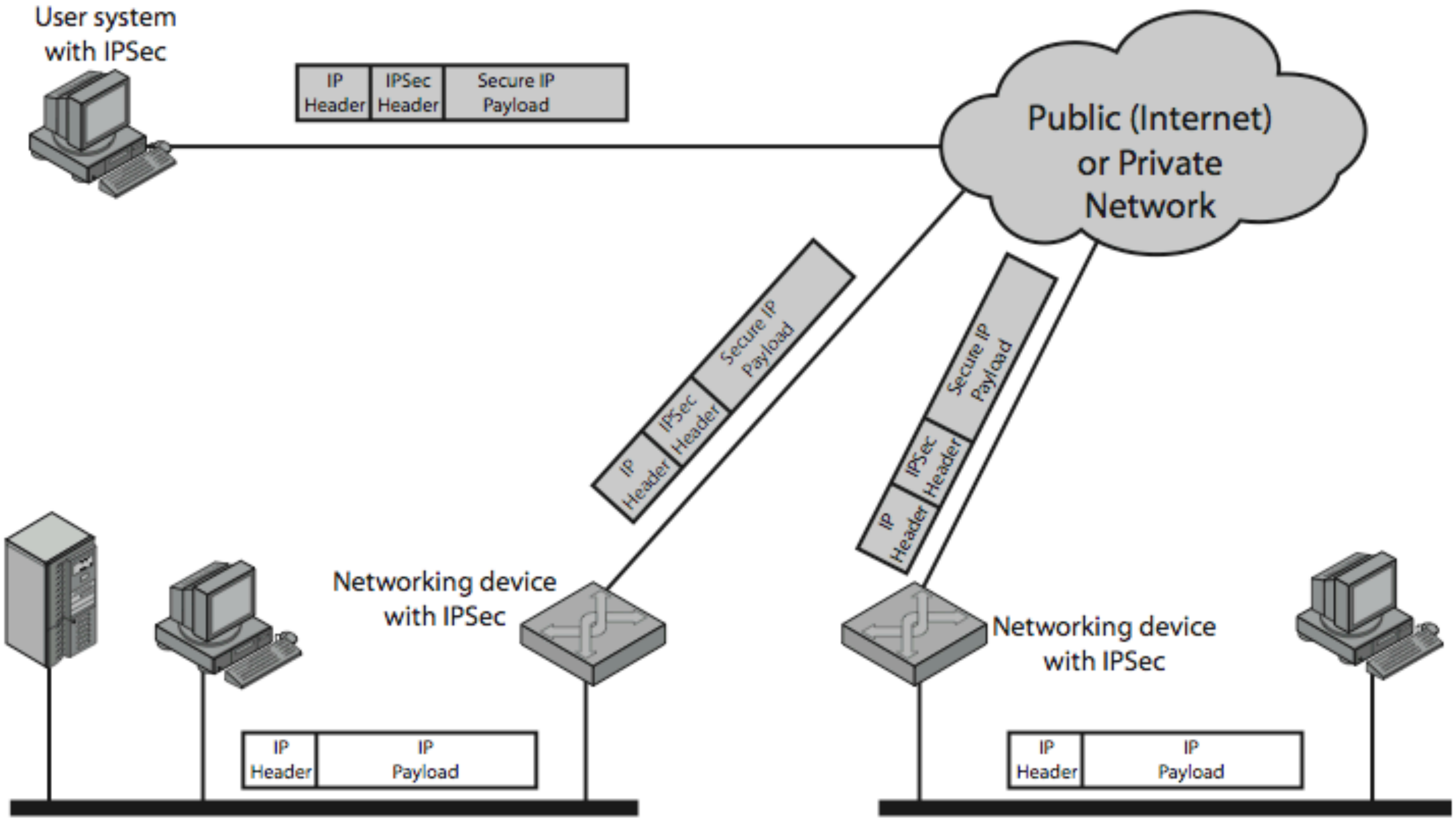
IP Security (IPSec)

- various application security mechanisms
  - eg. S/MIME, PGP, Kerberos, SSL/HTTPS
- security concerns cross protocol layers
- hence would like security implemented by the network for all applications
- authentication & encryption security features included in next-generation IPv6
- also usable in existing IPv4

IPSec

- general IP Security mechanisms
- provides
  - authentication
  - confidentiality
  - key management
- applicable to use over LANs, across public & private WANs, & for the Internet

IPSec Uses


User system with IPSec
IP Header
IPSec Header
Secure IP Payload
Public (Internet) or Private Network
IP Header
IPSec Header
Secure IP Payload
IP Header
IPSec Header
Secure IP Payload
Networking device with IPSec
Networking device with IPSec
IP Header
IP Payload
IP Header
IP Payload

Benefits of IPSec

- when implemented in a firewall/router, it provides strong security to all traffic crossing the perimeter
  - no overhead of security-related processing
- is below transport layer, hence transparent to applications
- can be transparent to end users; no need to train users on sec mechanisms such as keys
- secures routing architecture

IP Security Architecture

- mandatory in IPv6, optional in IPv4
- have three main functions:
  - Authentication Header (AH): auth only
  - Encapsulating Security Payload (ESP): auth & encryption
  - Key Exchange function: manual/auto exchange of keys
- Example: VPNs want both auth and encryption
  - hence usually use ESP
- specification is quite complex
  - numerous RFC’s 2401/2402/2406/2408

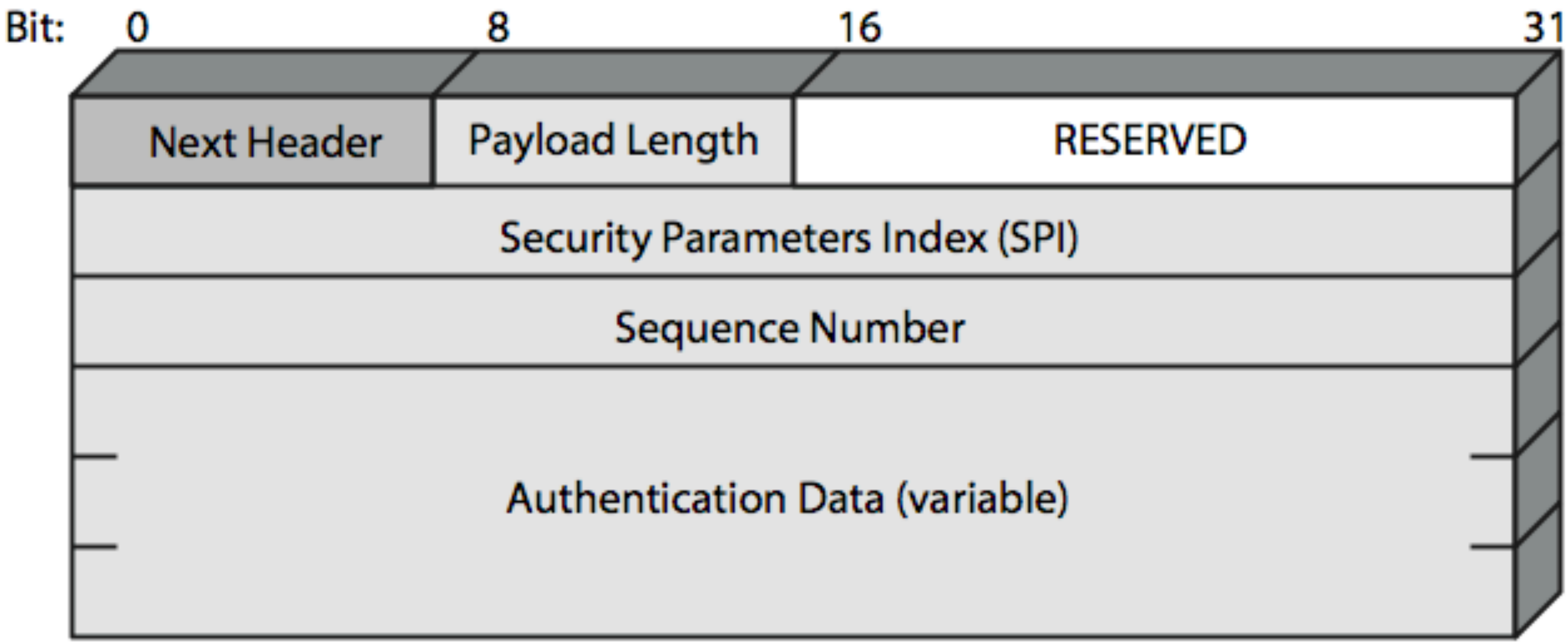
Security Associations (SA)

- a one-way relationship between sender & receiver that affords security for traffic flow
- defined by 3 parameters:
  - Security Parameters Index (SPI): and index in AH and ESP; tells receiver which SA to select
  - IP Destination Address: destination endpoint of a SA
  - Security Protocol Identifier
- has a number of other parameters
  - seq no, AH & EH info, lifetime etc

Authentication Header (AH)

- provides support for data integrity & authentication of IP packets
  - end system/router can authenticate user/app
  - prevents address spoofing attacks by tracking sequence numbers
- based on use of a MAC
  - HMAC-MD5-96 or HMAC-SHA-1-96
- parties must share a secret key

Authentication Header

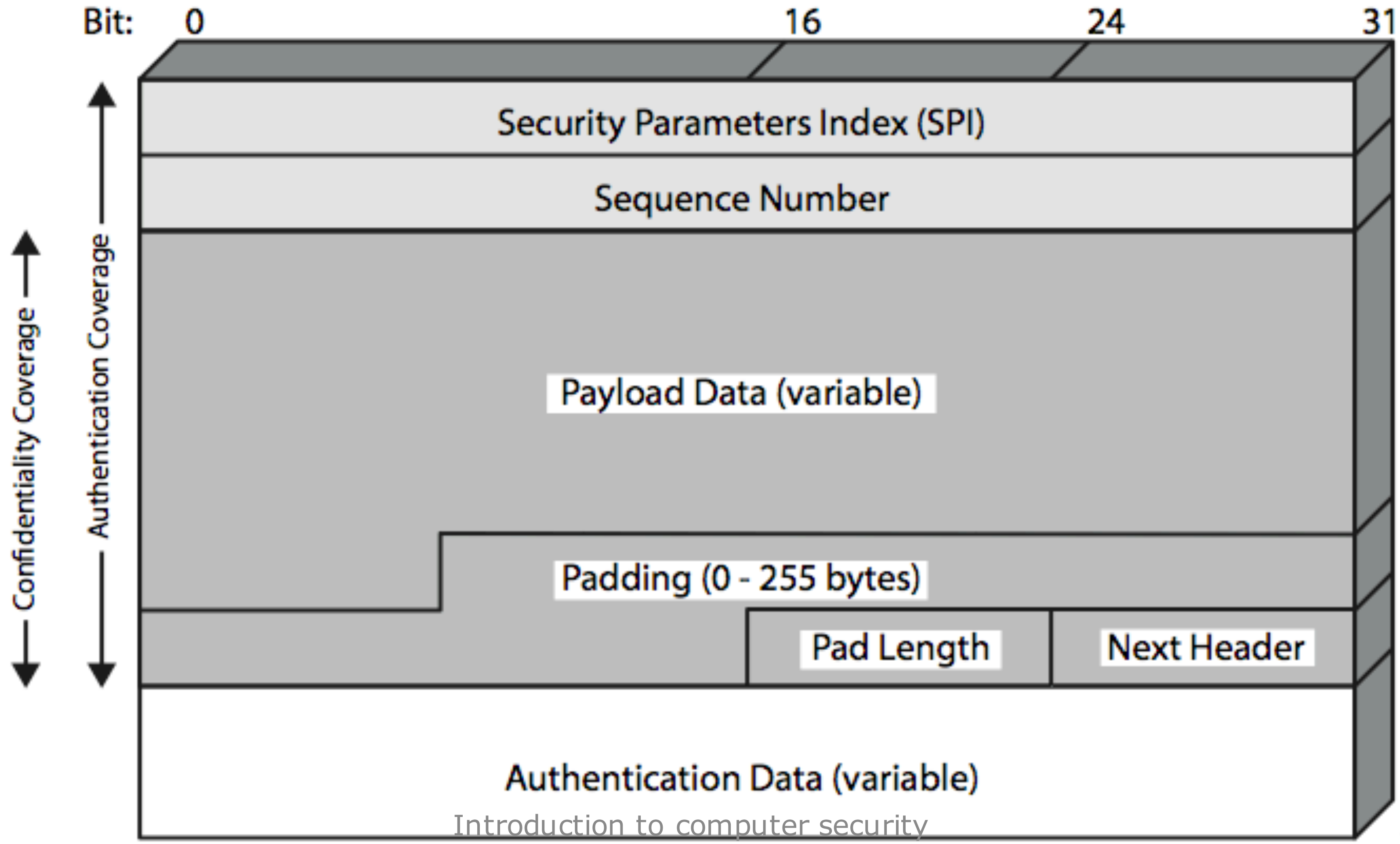


Bit: 0 8 16 31

Next Header	Payload Length	RESERVED
Security Parameters Index (SPI)		
Sequence Number		
Authentication Data (variable)		

Type of the header following this header; size of the AH; for future use; SA; increasing seq counter; authentication data, e.g., MAC or integrity check value

Encapsulating Security Payload (ESP)



Bit: 0 16 24 31

Security Parameters Index (SPI)

Sequence Number

Payload Data (variable)

Padding (0 - 255 bytes)

Pad Length

Next Header

Authentication Data (variable)

Confidentiality Coverage

Authentication Coverage

Key Management

- handles key generation & distribution
- typically need 2 pairs of keys
  - 2 per direction for AH & ESP
- manual key management
  - sysadmin manually configures every system
- automated key management
  - automated system for on demand creation of keys for SA’s in large systems
  - has Oakley & ISAKMP elements

Summary

- Secure Sockets Layer (SSL) / Transport Layer Security (TLS)
- IPv4 and IPv6 Security
- S/MIME (Secure/Multipurpose Internet Mail Extension)