Chapter 19

Electronic Mail Security

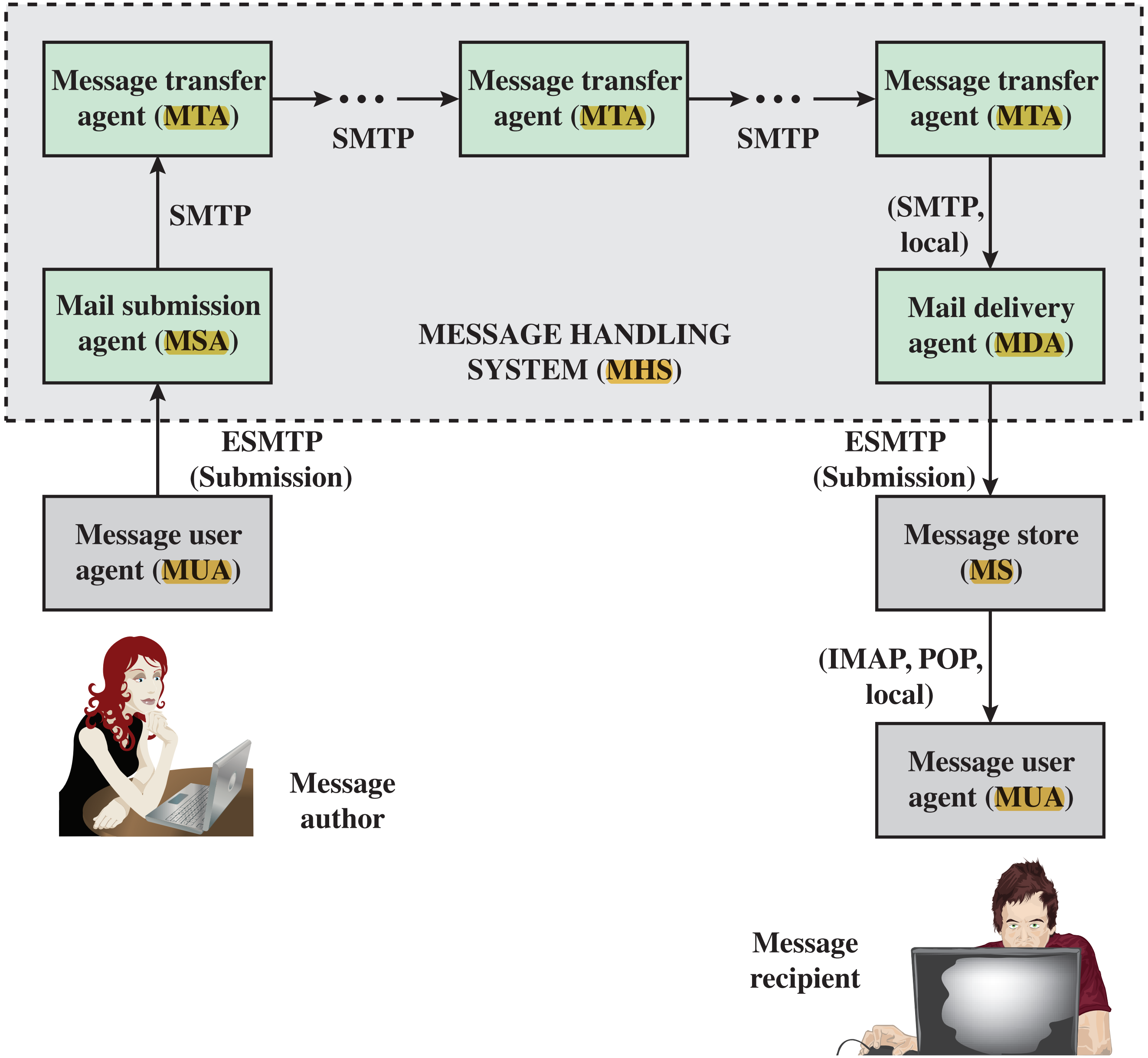


Figure 19.1 Function Modules and Standardized Protocols Used Between Them in the Internet Mail Architecture

Email Protocols

Two types of protocols are used for transferring email:

Used to move messages through the Internet from source to destination

- Simple Mail Transfer Protocol (SMTP)

Used to transfer messages between mail servers

- IMAP and POP are the most commonly used

SMTP

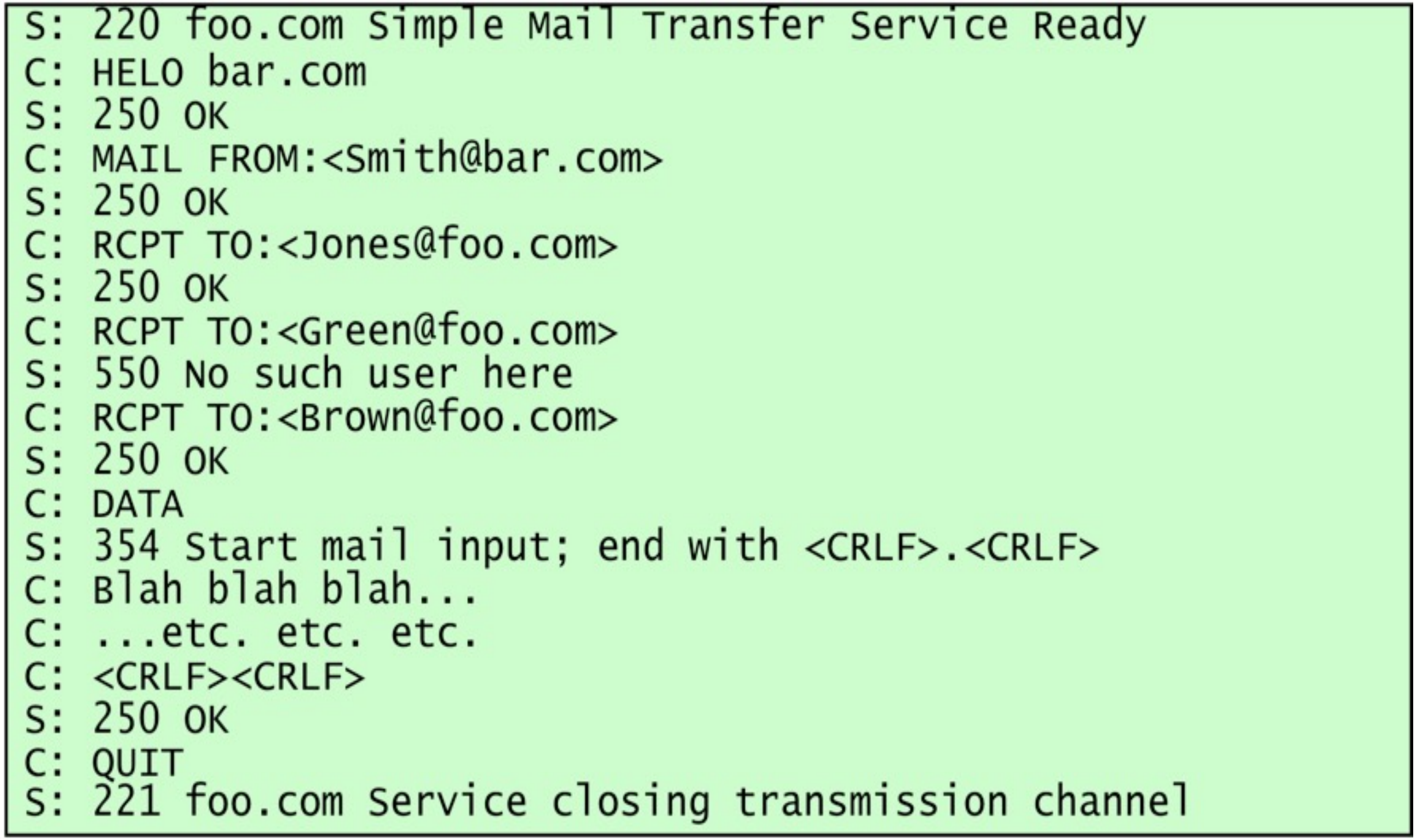
Simple Mail Transfer Protocol

Encapsulates an email message in an envelope and is used to relay the encapsulated messages from source to destination through multiple MTAs

Is a text-based client-server protocol

Was originally specified in 1982 as RFC 821

The term Extended SMTP (ESMTP) is often used to refer to later versions of SMTP



```
S: 220 foo.com Simple Mail Transfer Service Ready
C: HELO bar.com
S: 250 OK
C: MAIL FROM:<Smith@bar.com>
S: 250 OK
C: RCPT TO:<Jones@foo.com>
S: 250 OK
C: RCPT TO:<Green@foo.com>
S: 550 No such user here
C: RCPT TO:<Brown@foo.com>
S: 250 OK
C: DATA
S: 354 Start mail input; end with <CRLF>.<CRLF>
C: Blah blah blah...
C: ...etc. etc. etc.
C: <CRLF><CRLF>
S: 250 OK
C: QUIT
S: 221 foo.com Service closing transmission channel
```

Figure 19.2 Example SMTP Transaction Scenario

STARTTLS

- A significant security-related extension for SMTP
- Defined in RFC 3207
  - *SMTP Service Extension for Secure SMTP over Transport Layer Security,* February 2002
- Enables the addition of confidentiality and authentication in the exchange between SMTP agents
- This gives SMTP agents the ability to protect some or all of their communication from eavesdroppers and attackers

- Advantage of using STARTTLS is that the server can offer SMTP service on a single port, rather than requiring separate port numbers for secure and cleartext operations

Mail Access Protocols

POP3

- Post Office Protocol
- Allows an email client to download an email from an email server (MTA)
- POP3 user agents connect via TCP to the server
- After authorization, the UA can issue POP3 commands to retrieve and delete mail

IMAP

- Internet Mail Access Protocol
- Enables an email client to access mail on an email server
- Also uses TCP, with server TCP port 143
- Is more complex than POP3
- Provides stronger authentication and provides other functions not supported by POP3

RFC 5322

- Defines a format for text messages that are sent using electronic mail
- Messages are viewed as having an envelope and contents
  - The envelope contains whatever information is needed to accomplish transmission and delivery
  - The contents compose the object to be delivered to the recipient
  - RFC 5322 standard applies only to the contents
- The content standard includes a set of header fields that may be used by the mail system to create the envelope

Example Message

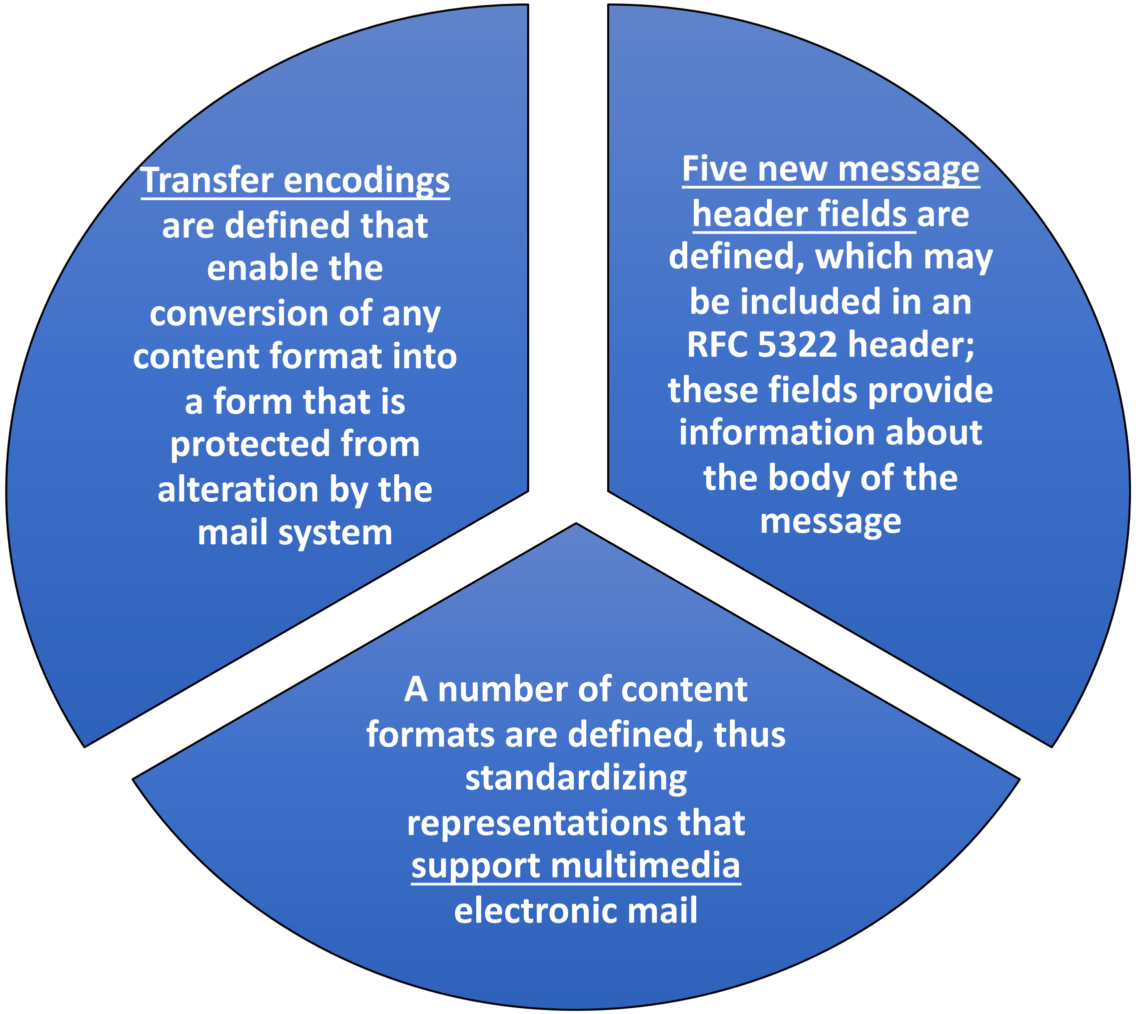
Date: October 8, 2009 2:15:49 PM EDT
From: “William Stallings” <ws@shore.net>
Subject: The Syntax in RFC 5322
To: Smith@Other-host.com
Cc: Jones@Yet-Another-Host.com

Hello. This section begins the actual message body, which is delimited from the message heading by a blank line.

Multipurpose Internet Mail Extensions (MIME)

- An extension to the RFC 5322 framework that is intended to address some of the problems and limitations of the use of Simple Mail Transfer Protocol (SMTP)
  - Is intended to resolve these problems in a manner that is compatible with existing RFC 5322 implementations
  - The specification is provided in RFCs 2045 through 2049

MIME specification includes the following elements:



Transfer encodings are defined that enable the conversion of any content format into a form that is protected from alteration by the mail system

Five new message header fields are defined, which may be included in an RFC 5322 header; these fields provide information about the body of the message

A number of content formats are defined, thus standardizing representations that support multimedia electronic mail

Limitations of the SMTP/5322 Scheme

- SMTP cannot transmit executable files or other binary objects
- SMTP cannot transmit text data that includes national language characters
- SMTP servers may reject mail message over a certain size
- SMTP gateways that translate between ASCII and the character code EBCDIC do not use a consistent set of mappings, resulting in translation problems
- SMTP gateways to X.400 electronic mail networks cannot handle nontextual data included in X.400 messages
- Some SMTP implementations do not adhere completely to the SMTP standards defined in RFC 821

MIME Specifications

- The MIME specification includes the following elements:
  - Five new message header fields are defined, which may be included in an RFC 5322 header
  - A number of content formats are defined, thus standardizing representations that support multimedia electronic mail
  - Transfer encodings are defined that enable the conversion of any content format into a form that is protected from alteration by the mail system

The Five Header Fields Defined in MIME

MIME-Version

- Must have the parameter value 1.0
- This field indicates that the message conforms to RFCs 2045 and 2046

Content-Type

- Describes the data contained in the body with sufficient detail that the receiving user agent can pick an appropriate agent or mechanism to represent the data to the user or otherwise deal with the data in an appropriate manner

Content-Transfer-Encoding

- Indicates the type of transformation that has been used to represent the body of the message in a way that is acceptable for mail transport

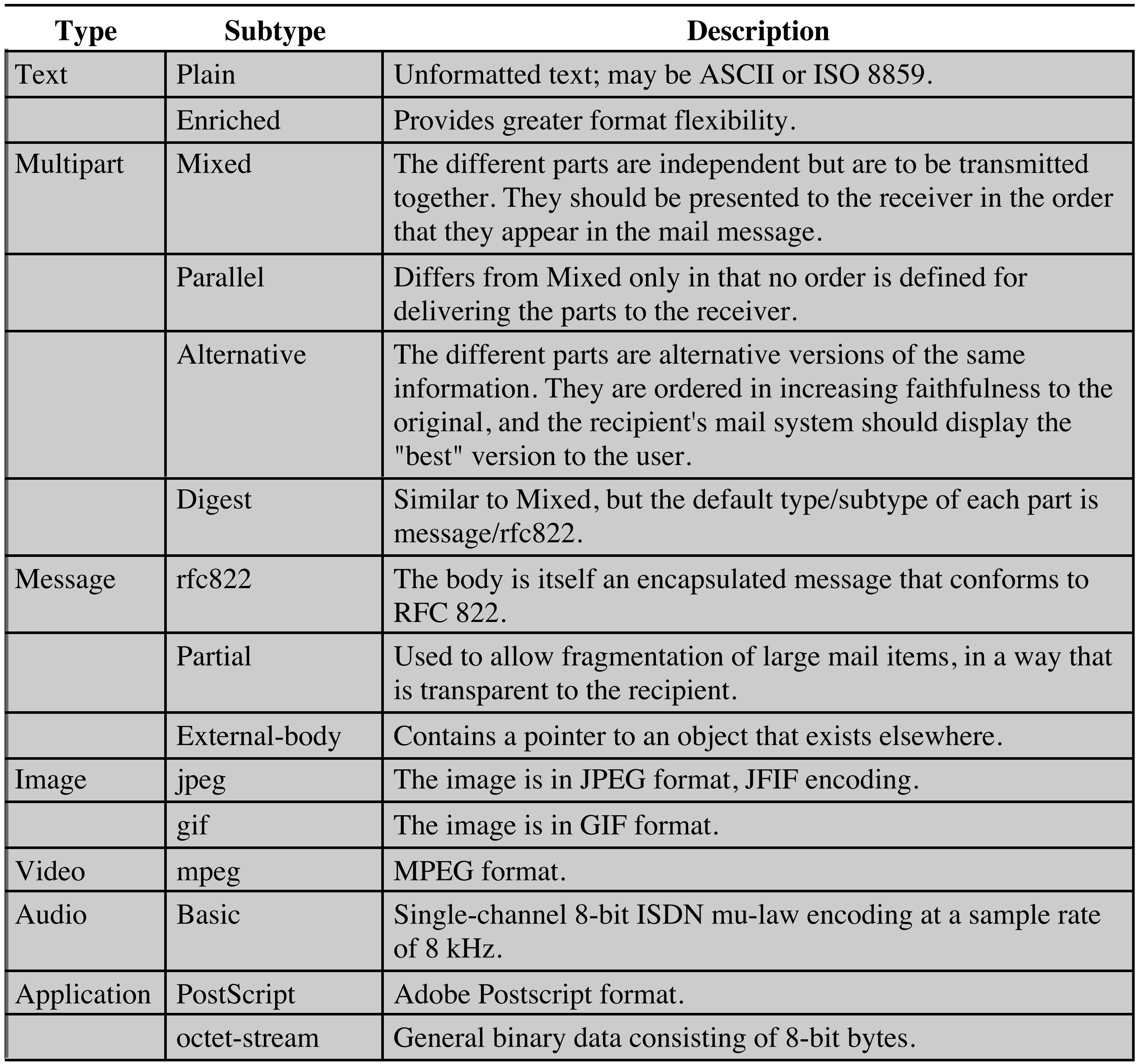
Content-ID

- Used to identify MIME entities uniquely in multiple contexts

Content-Description

- A text description of the object with the body; this is useful when the object is not readable

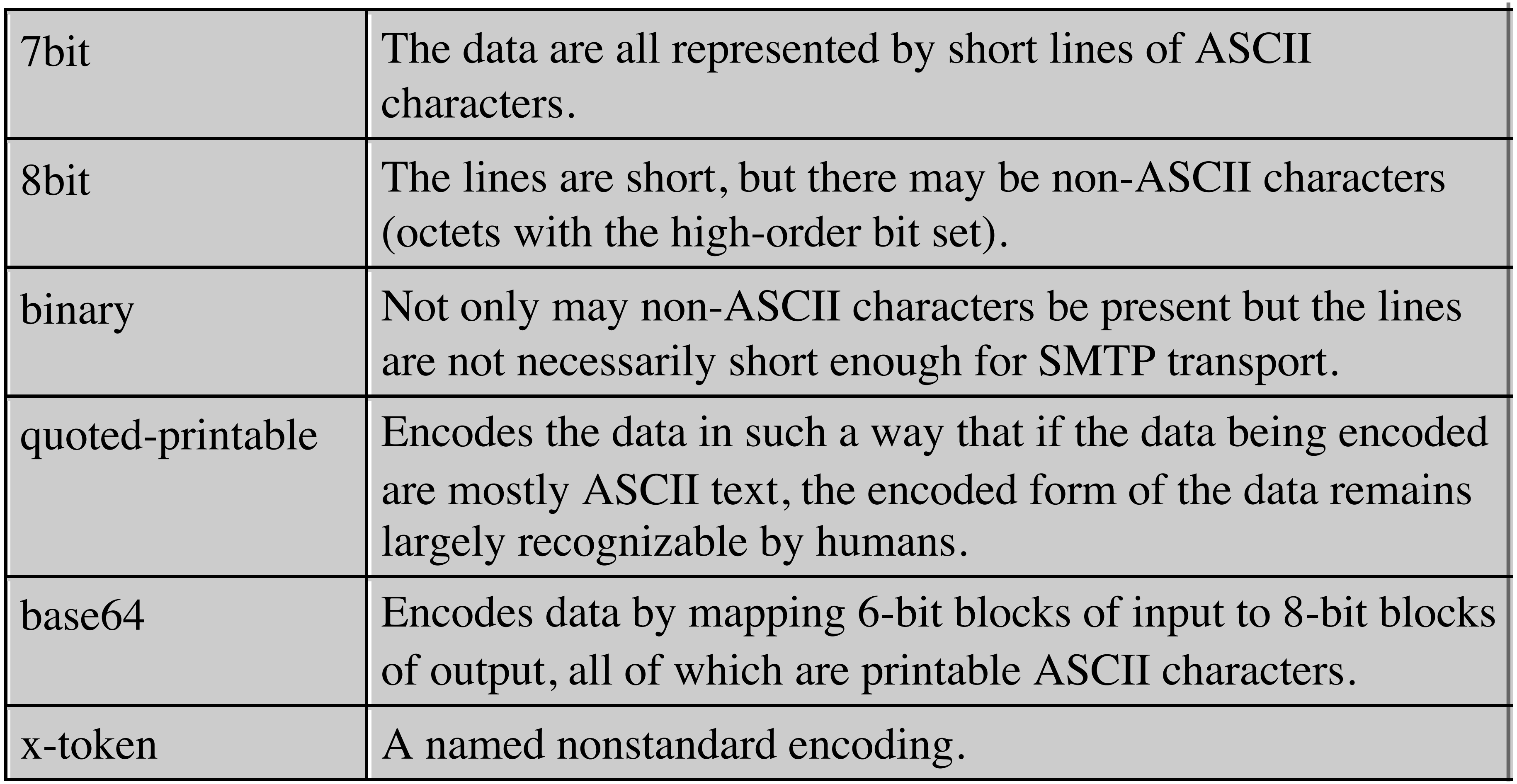
Table 19.1 MIME Content Types



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 format.
	octet-stream	General binary data consisting of 8-bit bytes.

(Table is on page 584 in the textbook)

Table 19.2 MIME Transfer Encodings



7bit	The data are all represented by short lines of ASCII characters.
8bit	The lines are short, but there may be non-ASCII characters (octets with the high-order bit set).
binary	Not only may non-ASCII characters be present but the lines are not necessarily short enough for SMTP transport.
quoted-printable	Encodes the data in such a way that if the data being encoded are mostly ASCII text, the encoded form of the data remains largely recognizable by humans.
base64	Encodes data by mapping 6-bit blocks of input to 8-bit blocks of output, all of which are printable ASCII characters.
x-token	A named nonstandard encoding.

(Table is on page 586 in the textbook)

```
MIME-Version: 1.0
   From: Nathaniel Borenstein <nsb@bellcore.com>
   To: Ned Freed <ned@innosoft.com>
   Subject: A multipart example
   Content-Type: multipart/mixed;
        boundary=unique-boundary-1
   This is the preamble area of a multipart message. Mail readers that understand multipart format should ignore this preamble.
    If you are reading this text, you might want to consider changing to a mail reader that understands how to properly display
    multipart messages.

    --unique-boundary-1
     ...Some text appears here...
   [Note that the preceding blank line means no header fields were given and this is text, with charset US ASCII.  It could have
    been done with explicit typing as in the next part.]

   --unique-boundary-1
   Content-type: text/plain; charset=US-ASCII
   This could have been part of the previous part, but illustrates explicit versus implicit typing of body parts.

   --unique-boundary-1
   Content-Type: multipart/parallel;   boundary=unique-boundary-2

   --unique-boundary-2
   Content-Type: audio/basic
   Content-Transfer-Encoding: base64
     ... base64-encoded 8000 Hz single-channel mu-law-format audio data goes here....

   --unique-boundary-2
   Content-Type: image/jpeg
   Content-Transfer-Encoding: base64
     ... base64-encoded image data goes here....

   --unique-boundary-2--
   --unique-boundary-1
   Content-type: text/enriched

   This is <bold><italic>richtext.</italic></bold> <smaller>as defined in RFC 1896</smaller>

    Isn't it <bigger><bigger>cool?</bigger></bigger>

   --unique-boundary-1
   Content-Type: message/rfc822

   From: (mailbox in US-ASCII)
   To: (address in US-ASCII)
   Subject: (subject in US-ASCII)
   Content-Type: Text/plain; charset=ISO-8859-1
   Content-Transfer-Encoding: Quoted-printable

     ... Additional text in ISO-8859-1 goes here ...

   --unique-boundary-1--
```

Figure 19.3   Example MIME Message Structure

Formats

Native Form

- The body to be transmitted is created in the system's native format
- The native character set is used and, where appropriate, local end-of-line conventions are used as well
- The body may be any format that corresponds to the local model for the representation of some form of information
- Examples include a UNIX-style text file, or a Sun raster image, or a VMS indexed file, and audio data in a system-dependent format stored only in memory

Canonical Form

- The entire body, including out-of-band information such as record lengths and possibly file attribute information, is converted to a universal canonical form
- The specific media type of the body as well as its associated attributes dictates the nature of the canonical form that is used
- Conversion to the proper canonical form may involve character set conversion, transformation of audio data, compression, or various other operations specific to the various media types

Email Security Threats

- Authenticity-related threats
  - Could result in unauthorized access to an enterprise's email system
- Integrity-related threats
  - Could result in unauthorized modification of email content
- Confidentiality-related threats
  - Could result in unauthorized disclosure of sensitive information
- Availability-related threats
  - Could prevent end users from being able to send or receive mail

Threat	Impact on Purported Sender	Impact on Receiver	Mitigation
Email sent by unauthorized MTA in enterprise (e.g. malware botnet)	Loss of reputation, valid email from enterprise may be blocked as possible spam/phishing attack.	UBE and/or email containing malicious links may be delivered into user inboxes	Deployment of domain-based authentication techniques. Use of digital signatures over email.
Email message sent using spoofed or unregistered sending domain	Loss of reputation, valid email from enterprise may be blocked as possible spam/phishing attack.	UBE and/or email containing malicious links may be delivered into user inboxes	Deployment of domain-based authentication techniques. Use of digital signatures over email.
Email message sent using forged sending address or email address (i.e. phishing, spear phishing)	Loss of reputation, valid email from enterprise may be blocked as possible spam/phishing attack.	UBE and/or email containing malicious links may be delivered. Users may inadvertently divulge sensitive information or PII.	Deployment of domain-based authentication techniques. Use of digital signatures over email.

Table 19.3
Email Threats and Mitigations

(Table can be found on page 587 in the textbook)

Threat	Impact on Purported Sender	Impact on Receiver	Mitigation
Email modified in transit	Leak of sensitive information or PII.	Leak of sensitive information, altered message may contain malicious information	Use of TLS to encrypt email transfer between server. Use of end-to-end email encryption.
Disclosure of sensitive information (e.g. PII) via monitoring and capturing of email traffic	Leak of sensitive information or PII.	Leak of sensitive information, altered message may contain malicious information	Use of TLS to encrypt email transfer between server. Use of end-to-end email encryption.
Unsolicited Bulk Email (i.e. spam)	None, unless purported sender is spoofed.	UBE and/or email containing malicious links may be delivered into user inboxes	Techniques to address UBE.
DoS/DDoS attack against an enterprises’ email servers	Inability to send email.	Inability to receive email.	Multiple mail servers, use of cloud-based email providers.

Table 19.3

Email Threats and Mitigations

(Table can be found on page 587 in the textbook)

Counter Threat Protocols

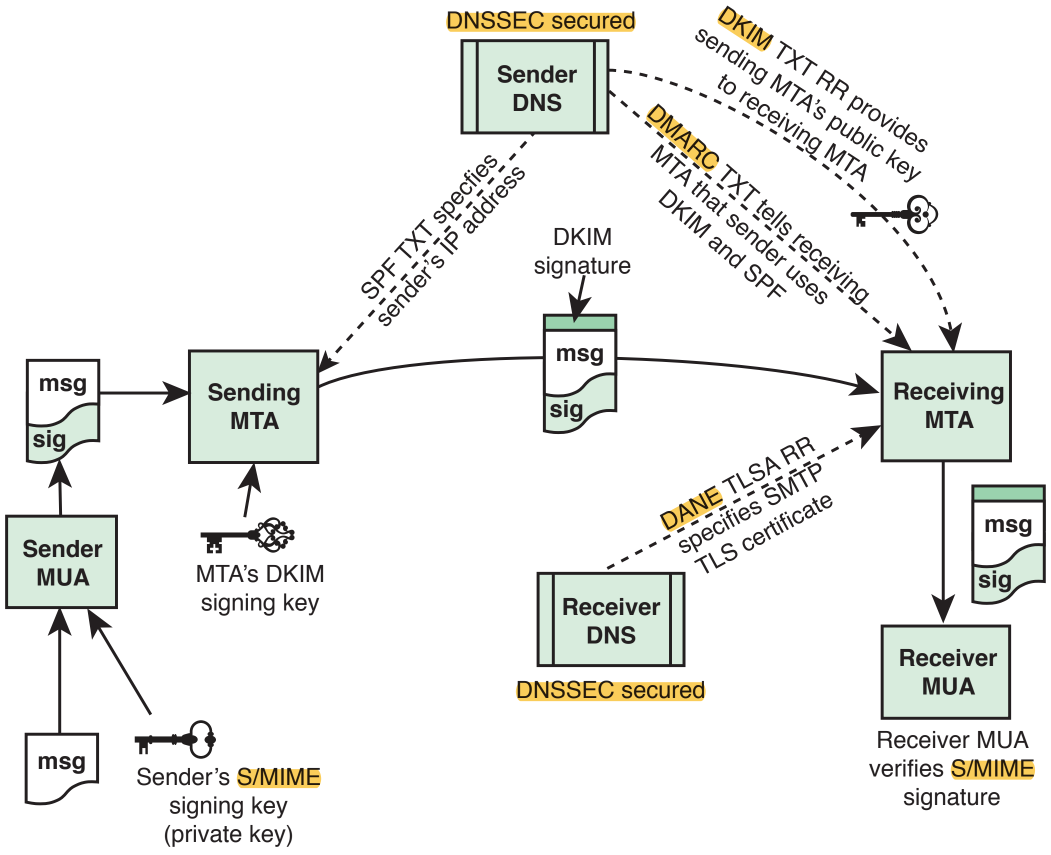
- SP800-177 recommends use of a variety of standardized protocols as a means for countering threats:
  - **STARTTLS**
    - An SMPT security extension that provides authentication, integrity, non-repudiation and confidentiality for the entire SMTP message by running SMTP over TLS
  - **S/MIME**
    - Provides authentication, integrity, non-repudiation and confidentiality of the message body carried in SMTP messages
  - DNS Security Extensions (**DNSSEC**)
    - Provides authentication and integrity protection of DNS data, and is an underlying tool used by various email security protocols

DNS PKI

  - DNS-based Authentication of Named Entities (**DANE**)
    - Is designed to overcome problems in the certificate authority (CA) system by providing an alternative channel for authenticating public keys based on DNSSEC, with the result that the same trust relationships used to certify IP addresses are used to certify servers operating on those addresses

Counter Threat Protocols

- SP800-177 recommends use of a variety of standardized protocols as a means for countering threats:
  - Sender Policy Framework (**SPF**)
    - Uses the Domain Name System (DNS) to allow domain owners to create records that associate the domain name with a specific IP address range of authorized message senders.
  - DomainKeys Identified Mail (**DKIM**)
    - Enables an MTA to sign selected headers and the body of a message. This validates the source domain of the mail and provides message body integrity
  - Domain-based Message Authentication, Reporting, and Conformance (**DMARC**)
    - Lets senders know the proportionate effectiveness of their SPF and DKIM policies, and signals to receivers what action should be taken in various individual and bulk attack scenarios


DNSSEC secured
Sender DNS
DKIM TXT RR provides sending MTA's public key to receiving MTA
DMARC TXT tells receiving MTA that sender uses DKIM and SPF
SPF TXT specfies sender's IP address
DKIM signature
msg
sig
Sending MTA
Receiving MTA
msg
sig
MTA's DKIM signing key
DANE TLSA RR specifies SMTP TLS certificate
Receiver DNS
DNSSEC secured
msg
sig
Sender MUA
msg
Sender's S/MIME signing key (private key)
msg
sig
Receiver MUA
Receiver MUA verifies S/MIME signature

DANE = DNS-based Authentication of Named Entities
DKIM = DomainKeys Identified Mail
DMARC = Domain-based Message Authentication, Reporting, and Conformance
DNSSEC = Domain Name System Security Extensions
SPF = Sender Policy Framework
S/MIME = Secure Multi-Purpose Internet Mail Extensions
TLSA RR = Transport Layer Security Authentication Resource Record

Figure 19.2 The Interrelationship of DNSSEC, SPF, DKIM, DMARC, DANE, and S/MIME for Assuring Message Authenticity and Integrity

Secure/Multipurpose Internet Mail Extension (S/MIME)

- A security enhancement to the MIME Internet e-mail format standard based on technology from RSA Data Security
- The most important documents relevant to S/MIME include:
  - RFC 5750, S/MIME Version 3.2 Certificate Handling
  - RFC 5751, S/MIME Version 3.2 Message Specification
  - RFC 4134, Examples of S/MIME Messages
  - RFC 2634, Enhanced Security Services for S/MIME
  - RFC 5652, Cryptographic Message Syntax (CMS)
  - RFC 3370, CMS Algorithms
  - RFC 5752, Multiple Signatures in CMS
  - RFC 1847, Security Multiparts for MIME – Multipart/Signed and Multipart/Encrypted

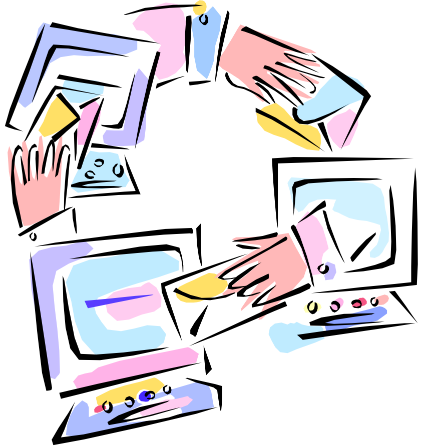


Table 19.4 Summary of S/MIME Services

Function	Typical Algorithm	Typical Action
Digital signature	RSA/SHA-256	A hash code of a message is created using SHA-256. This message digest is encrypted using SHA-256 with the sender's private key and included with the message.
Message encryption	AES-128 with CBC	A message is encrypted using AES-128 with CBC with a one-time session key generated by the sender. The session key is encrypted using RSA with the recipient's public key and included with the message.
Compression	unspecified	A message may be compressed for storage or transmission.
E-mail compatibility	Radix-64 conversion	To provide transparency for e-mail applications, an encrypted message may be converted to an ASCII string using radix-64 conversion.

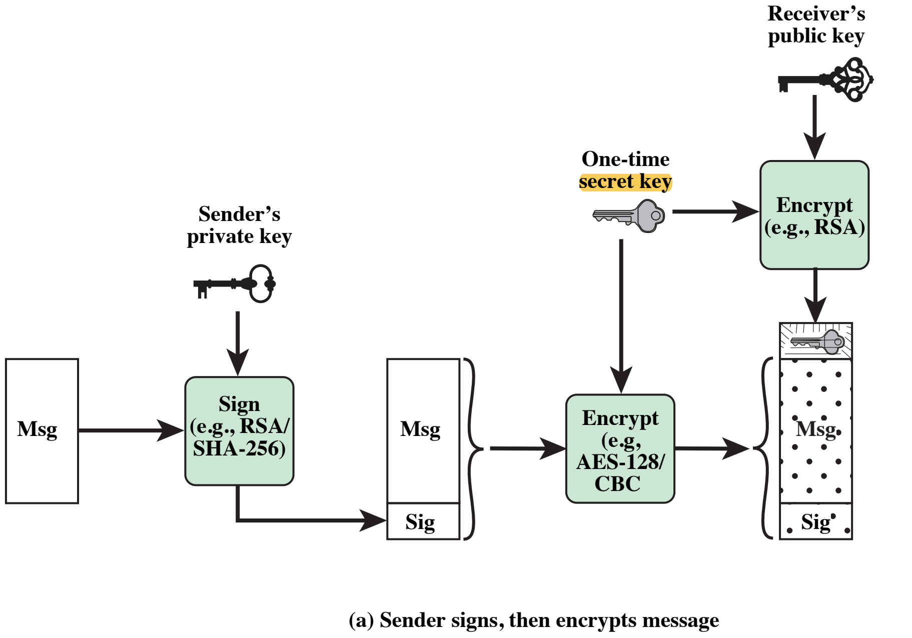
(Table is on page 590 in the textbook)

Authentication

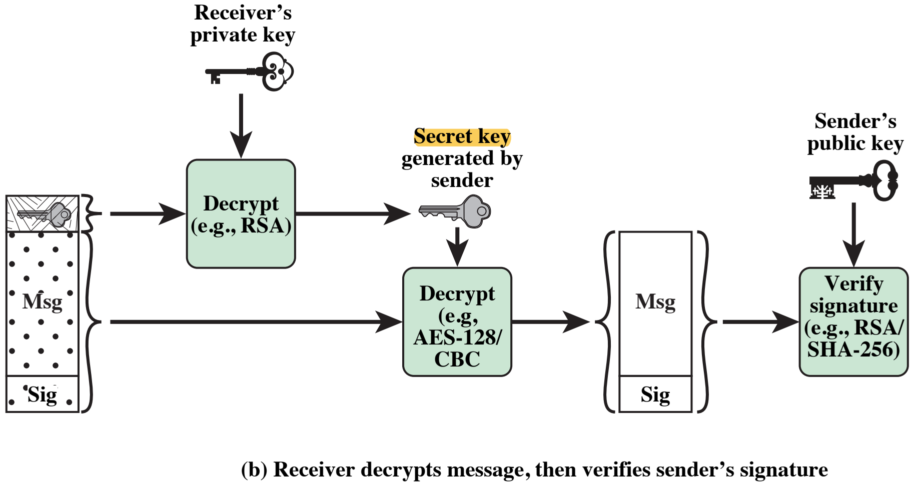
- Provided by means of a digital signature
  - The sender creates a message
  - SHA-256 is used to generate a 256-bit message digest of the message
  - The message digest is encrypted with RSA using the sender's private key, and the result is appended to the message. Also appended is identifying information for the signer, which will enable the receiver to retrieve the signer's public key
  - The receiver uses RSA with the sender's public key to decrypt and recover the message digest
  - The receiver generates a new message digest for the message and compares it with the decrypted hash code. If the two match, the message is accepted as authentic
- Detached signatures are supported
  - A detached signature may be stored and transmitted separately from the message it signs

Confidentiality

- S/MIME provides confidentiality by encrypting messages
  - Most commonly AES with a 128-bit key is used, with the cipher block chaining (CBC) mode
- The key itself is also encrypted, typically with RSA
- Each symmetric key, referred to as a content-encryption key, is used <u>only once</u>
  - A new key is generated as a random number for each message
  - Because it is to be used only once, the content-encryption key is bound to the message and transmitted with it
  - To protect the key, it is encrypted with the receiver’s public key
  - To reduce encryption time, the combination of symmetric and public-key encryption is used
  - Only the recipient is able to recover the session key that is bound to the message



(a) Sender signs, then encrypts message



(b) Receiver decrypts message, then verifies sender's signature

Figure 19.3 Simplified S/MIME Functional Flow

E-mail Compatibility

- Many electronic mail systems only permit the use of blocks consisting of ASCII text
  - To accommodate this restriction, S/MIME provides the service of converting the raw 8-bit binary stream to a stream of printable ASCII characters
  - The scheme used for this purpose is Base-64 conversion
    - Each group of three octets of binary data is mapped into four ASCII characters
    - The Base64 algorithm blindly converts the input stream to Base64 format regardless of content, even if the input happens to be ASCII text

- RFC 5751 recommends that even if outer 7-bit encoding is not used, the original MIME content should be 7-bit encoded

Compression

- S/MIME offers the ability to compress a message
- This has the benefit of saving space both for email transmission and for file storage
- Compression can be applied in any order with respect to the signing and message encryption operations
- RFC 5751 provides these guidelines:
  - Compression of binary encoded encrypted data is discouraged, since it will not yield significant compression; Base64 encrypted data could very well benefit, however
  - If a lossy compression algorithm is used with signing, you will need to compress first, then sign

S/MIME Message Content Types

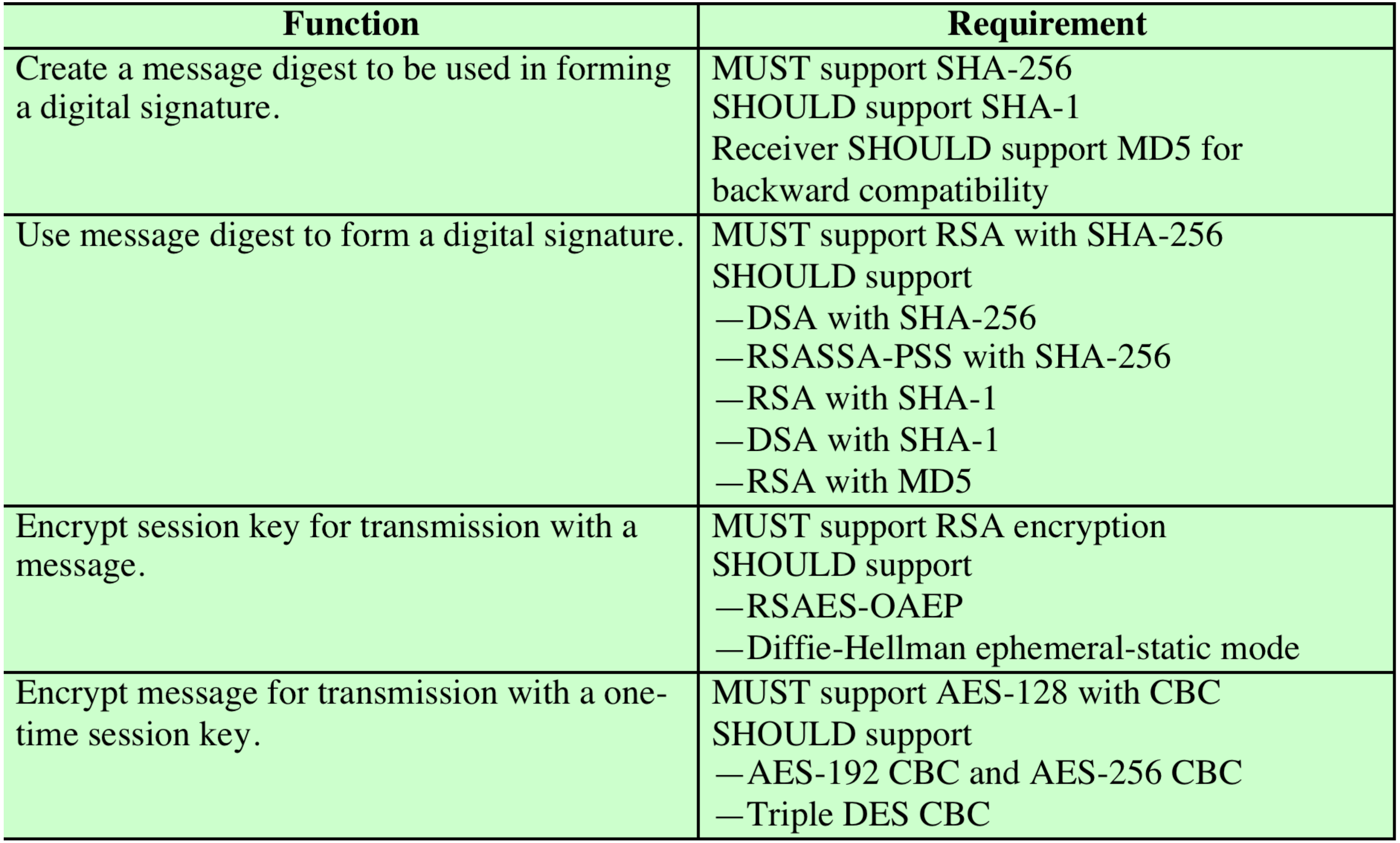
- Defined in RFC 5652, Cryptographic Message Syntax
  - Data
    - Refers to the inner MIME-encoded message content, which may then be encapsulated in a SignedData, EnvelopedData, or CompressedData content type
  - SignedData
    - Used to apply a digital signature to a message
  - EnvelopedData
    - This consists of encrypted content of any type and encrypted content encryption keys for one or more recipients
  - CompressedData
    - Used to apply data compression to a message
- **Clear signing**
  - A digital signature is calculated for a MIME-encoded message and the two parts, the message and signature, form a multipart MIME message
  - Can be read and their signatures verified by email entities that do not implement S/MIME

Securing a MIME Entity

- S/MIME secures a MIME entity with a signature, encryption, or both
- The MIME entity is prepared according to the normal rules for MIME message preparation
  - The MIME entity plus some security-related data, such as algorithm identifiers and certificates, are processed by S/MIME to produce what is known as a PKCS object
  - A PKCS object is then treated as message content and wrapped in MIME
- In all cases the message to be sent is converted to canonical form

Table 19.5

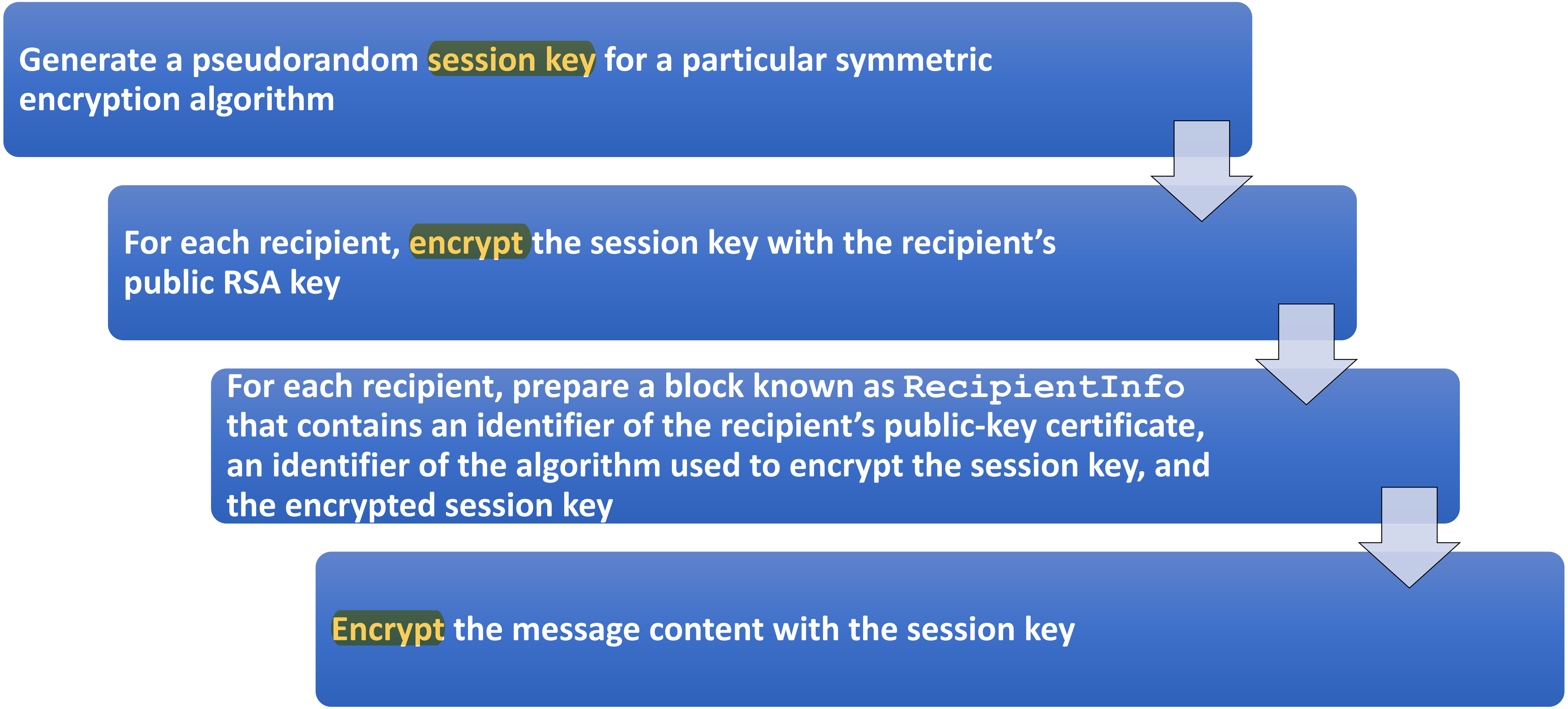
Cryptographic Algorithms Used in S/MIME



Function	Requirement
Create a message digest to be used in forming a digital signature.	MUST support SHA-256 SHOULD support SHA-1 Receiver SHOULD support MD5 for backward compatibility
Use message digest to form a digital signature.	MUST support RSA with SHA-256 SHOULD support —DSA with SHA-256 —RSASSA-PSS with SHA-256 —RSA with SHA-1 —DSA with SHA-1 —RSA with MD5
Encrypt session key for transmission with a message.	MUST support RSA encryption SHOULD support —RSAES-OAEP —Diffie-Hellman ephemeral-static mode
Encrypt message for transmission with a one-time session key.	MUST support AES-128 with CBC SHOULD support —AES-192 CBC and AES-256 CBC —Triple DES CBC

EnvelopedData

- The steps for preparing an envelopedData MIME are:



Generate a pseudorandom session key for a particular symmetric encryption algorithm

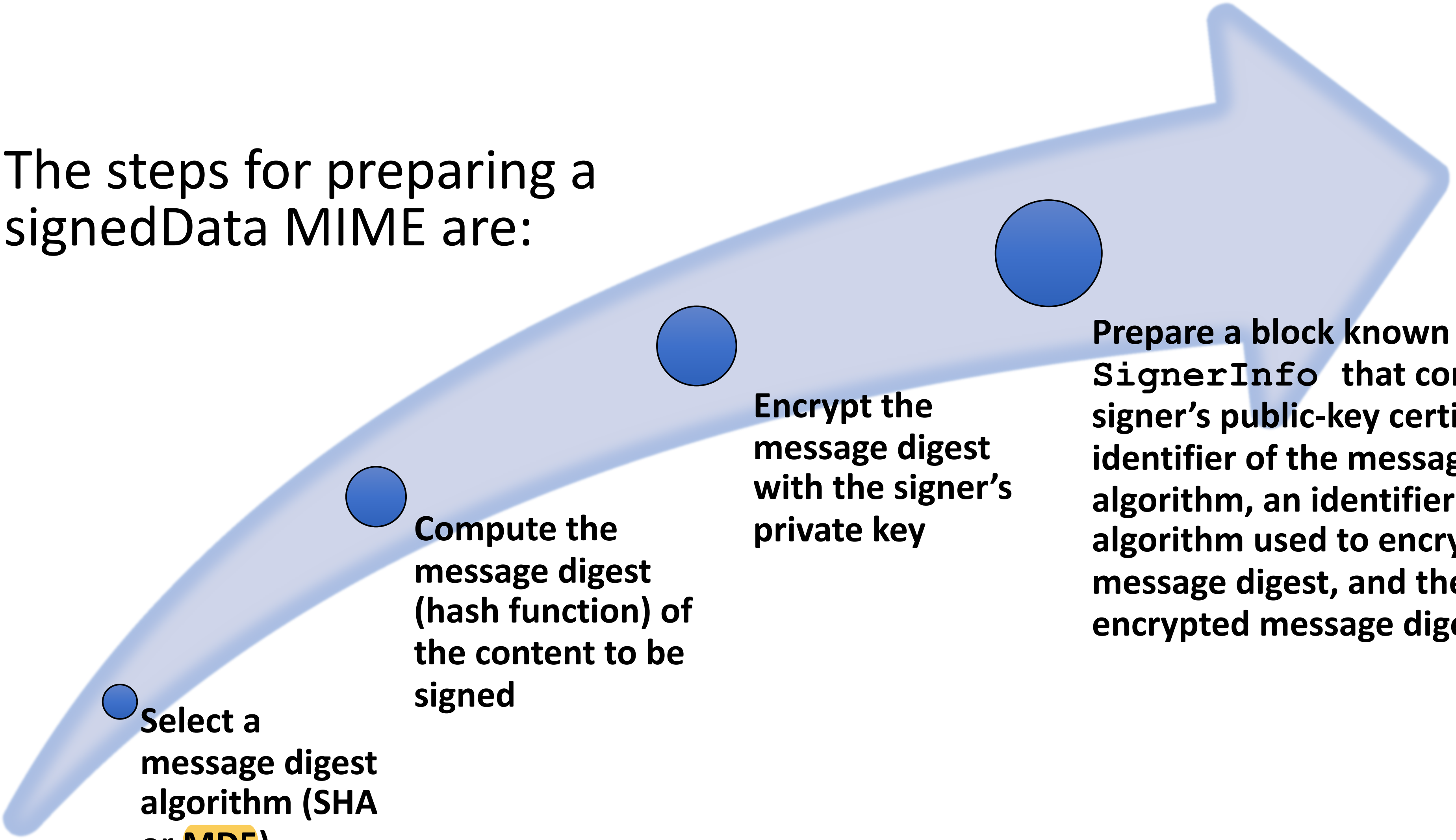
For each recipient, encrypt the session key with the recipient's public RSA key

For each recipient, prepare a block known as `RecipientInfo` that contains an identifier of the recipient's public-key certificate, an identifier of the algorithm used to encrypt the session key, and the encrypted session key

Encrypt the message content with the session key

SignedData

- The steps for preparing a signedData MIME are:



Select a message digest algorithm (SHA or MD5)

Compute the message digest (hash function) of the content to be signed

Encrypt the message digest with the signer’s private key

Prepare a block known as `SignerInfo` that contains the signer’s public-key certificate, an identifier of the message digest algorithm, an identifier of the algorithm used to encrypt the message digest, and the encrypted message digest

Clear Signing

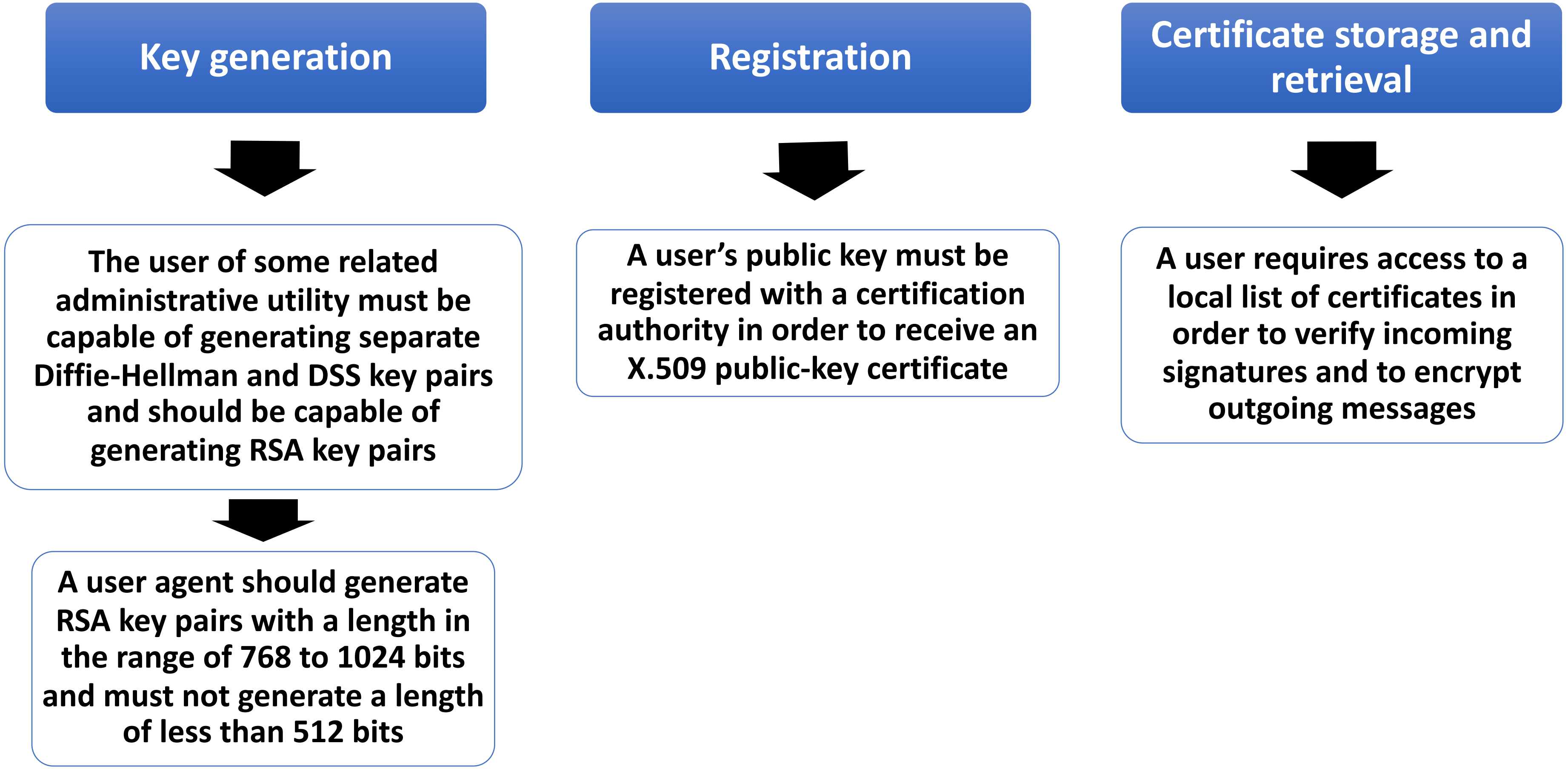
- Achieved using the multipart content type with a signed subtype
- This signing process does not involve transforming the message to be signed
- Recipients with MIME capability but <u>not</u> S/MIME capability are able to read the incoming message

S/MIME Certificate Processing

- S/MIME uses public-key certificates that conform to version 3 of X.509
- S/MIME managers and/or users must configure each client with a list of trusted keys and with certificate revocation lists
  - The responsibility is local for maintaining the certificates needed to verify incoming signatures and to encrypt outgoing messages
- The certificates are signed by certification authorities

User Agent Role

- An S/MIME user has several key-management functions to perform:



Key generation

The user of some related administrative utility must be capable of generating separate Diffie-Hellman and DSS key pairs and should be capable of generating RSA key pairs

A user agent should generate RSA key pairs with a length in the range of 768 to 1024 bits and must not generate a length of less than 512 bits

Registration

A user's public key must be registered with a certification authority in order to receive an X.509 public-key certificate

Certificate storage and retrieval

A user requires access to a local list of certificates in order to verify incoming signatures and to encrypt outgoing messages

Enhanced Security Services

- RFC 2634 defines four enhanced security services for S/MIME:
  - Signed receipt
    - Returning a signed receipt provides proof of delivery to the originator of a message and allows the originator to demonstrate to a third party that the recipient received the message
  - Security labels
    - A set of security information regarding the sensitivity of the content that is protected by S/MIME encapsulation
  - Secure mailing lists
    - An S/MIME Mail List Agent (MLA) can take a single incoming message, perform the recipient-specific encryption for each recipient, and forward the message
  - Signing certificates
    - This service is used to securely bind a sender’s certificate to their signature through a signing certificate attribute

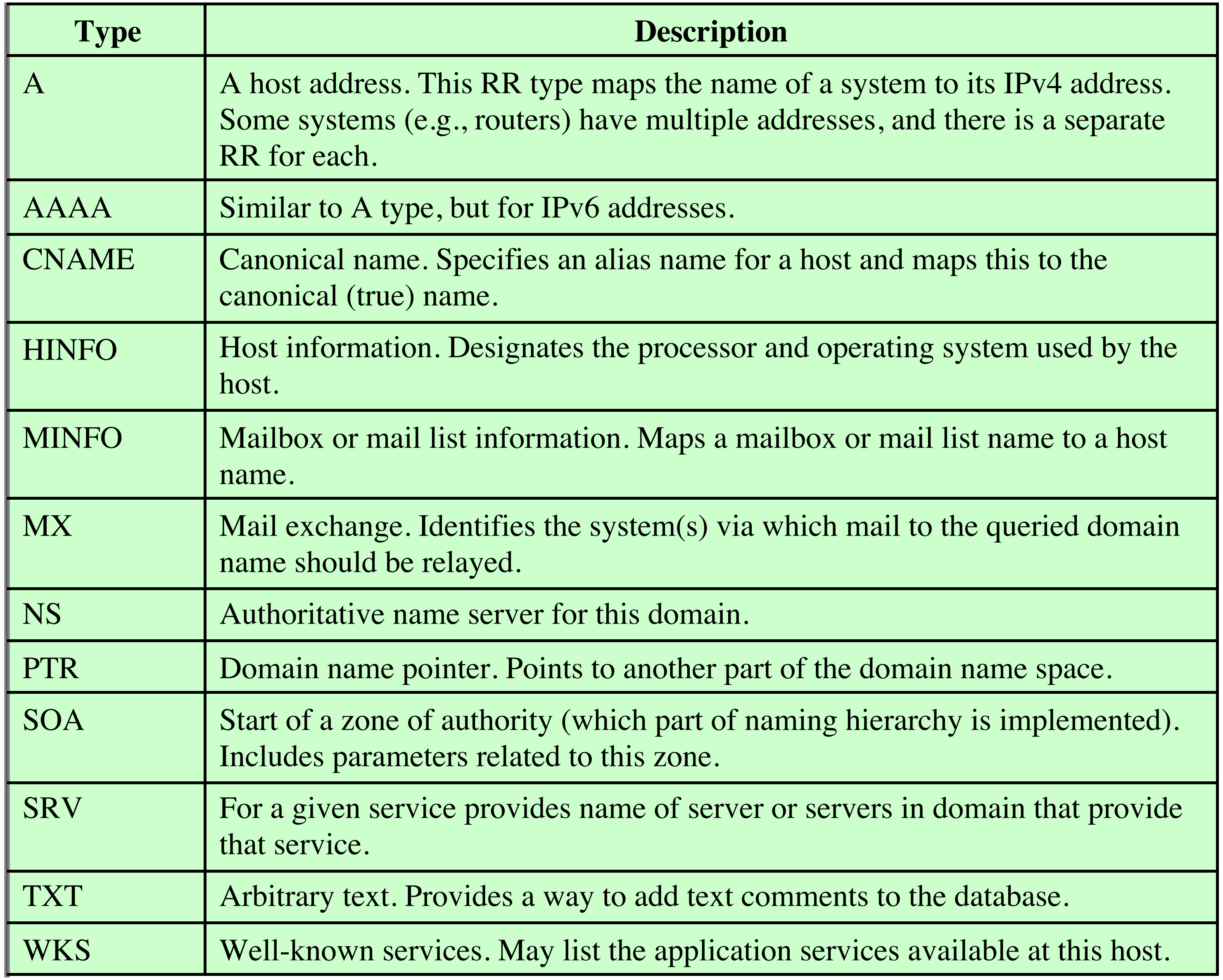
Domain Name System (DNS)

- A directory lookup service that provides a mapping between the name of a host on the Internet and its numeric IP address
- Is essential to the functioning of the Internet
- Is used by MUAs and MTAs to find the address of the next hop server for mail delivery
- Is comprised of four elements:
  - Domain name space
  - DNS database
  - Name servers
  - Resolvers

DNS Database

- DNS is based on a hierarchical database containing resource records (RRs) that include the name, IP address, and other information about hosts
- The key features of the database are:
  - Variable-depth hierarchy for names
  - Distributed database
  - Distribution controlled by the database
  - Using this database, DNS servers provide a name-to-address directory service for network applications that need to locate specific servers

Table 19.5 Resource Record Types



Type	Description
A	A host address. This RR type maps the name of a system to its IPv4 address. Some systems (e.g., routers) have multiple addresses, and there is a separate RR for each.
AAAA	Similar to A type, but for IPv6 addresses.
CNAME	Canonical name. Specifies an alias name for a host and maps this to the canonical (true) name.
HINFO	Host information. Designates the processor and operating system used by the host.
MINFO	Mailbox or mail list information. Maps a mailbox or mail list name to a host name.
MX	Mail exchange. Identifies the system(s) via which mail to the queried domain name should be relayed.
NS	Authoritative name server for this domain.
PTR	Domain name pointer. Points to another part of the domain name space.
SOA	Start of a zone of authority (which part of naming hierarchy is implemented). Includes parameters related to this zone.
SRV	For a given service provides name of server or servers in domain that provide that service.
TXT	Arbitrary text. Provides a way to add text comments to the database.
WKS	Well-known services. May list the application services available at this host.

(Table is on page 600 in the textbook)

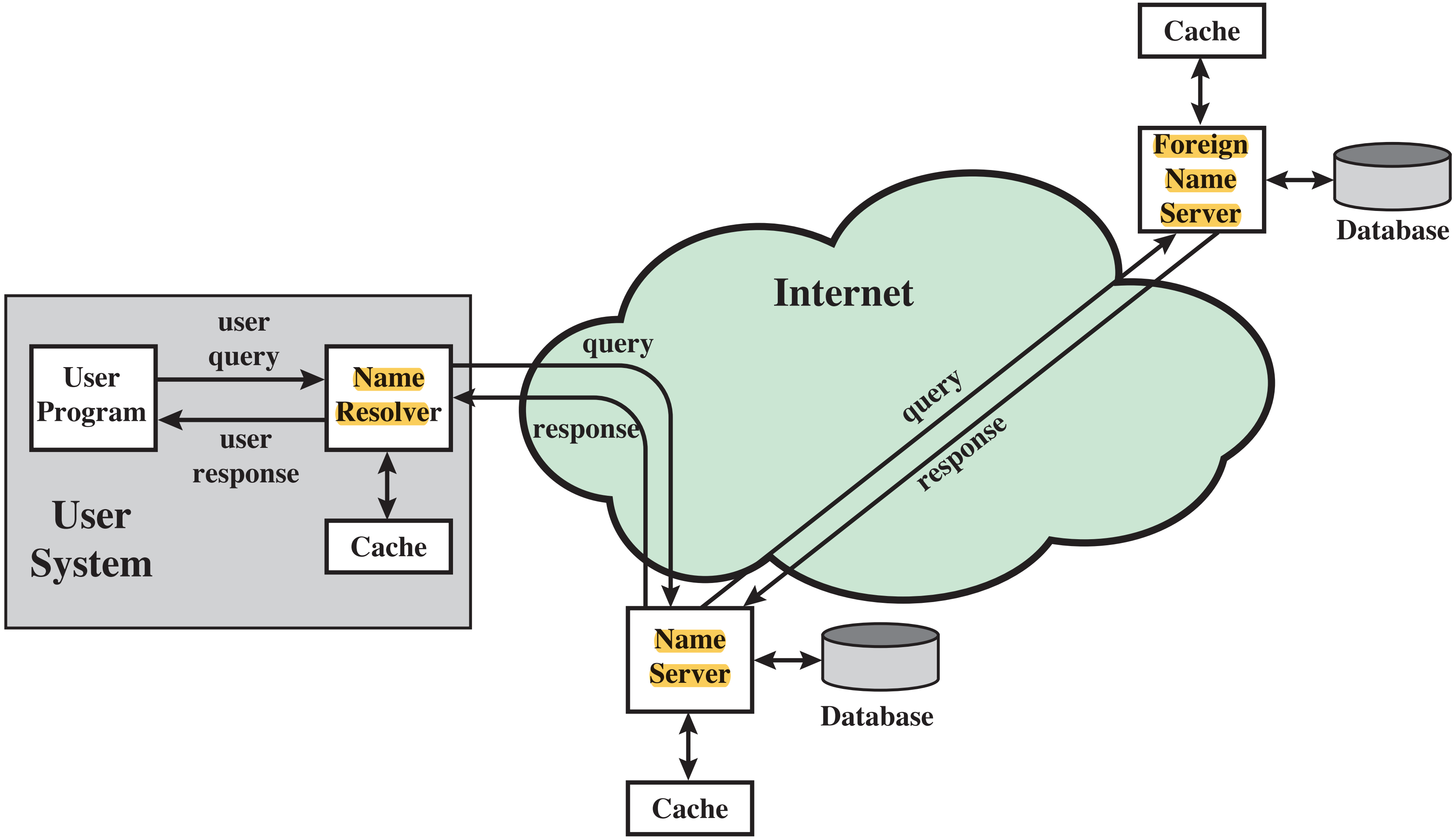


Figure 19.4 DNS Name Resolution

DNSSEC

- DNS Security Extensions
- Provides end-to-end protection through the use of digital signatures that are created by responding zone administrators and verified by a recipient's resolver software
- Avoids the need to trust intermediate name servers and resolvers that cache or route the DNS records originating from the responding zone administrator before they reach the source of the query
- Consists of a set of new resource record types and modifications to the existing DNS protocol
- Defined in these documents:
    - RFC 4033, DNS Security Introduction and Requirements
    - RFC 4034, Resource Records for the DNS Security Extensions
    - RFC 4035, Protocol Modifications for the DNS Security Extensions

DNSSEC Operation

- In essence, DNSSEC is designed to protect DNS clients from accepting forged or altered DNS resource records
- It does this by using digital signatures to provide:
  - Data origin authentication
    - Ensures that data has originated from the correct source
  - Data integrity verification
    - Ensures that the content of a RR has not been modified

- Trust in the public key of the source is established by starting from a trusted zone and establishing the chain of trust down to the current source of response through successive verifications of signature of the public key of a child by its parent
  - The public key of the trusted zone is called the ***trust anchor***

Resource Records for DNSSEC

- RFC 4034 defines four new DNS resource records:
  - **DNSKEY**
    - Contains a public key
  - **RRSIG**
    - A resource record digital signature
  - **NSEC**
    - Authenticated denial of existence record
  - **DS**
    - Delegation signer
- DNSSEC depends on establishing the authenticity of the DNS hierarchy leading to the domain name in question, and thus its operation depends on beginning the use of cryptographic digital signatures in the root zone
- The DS resource record facilitates key signing and authentication between DNS zones to create an authentication chain from the root of the DNS tree down to a specific domain name
- To secure all DNS lookups DNSSEC uses the NSEC resource record to authenticate negative responses to queries
- NSEC is used to identify the range of DNS names or resource record types that do not exist among the sequence of domain names in a zone

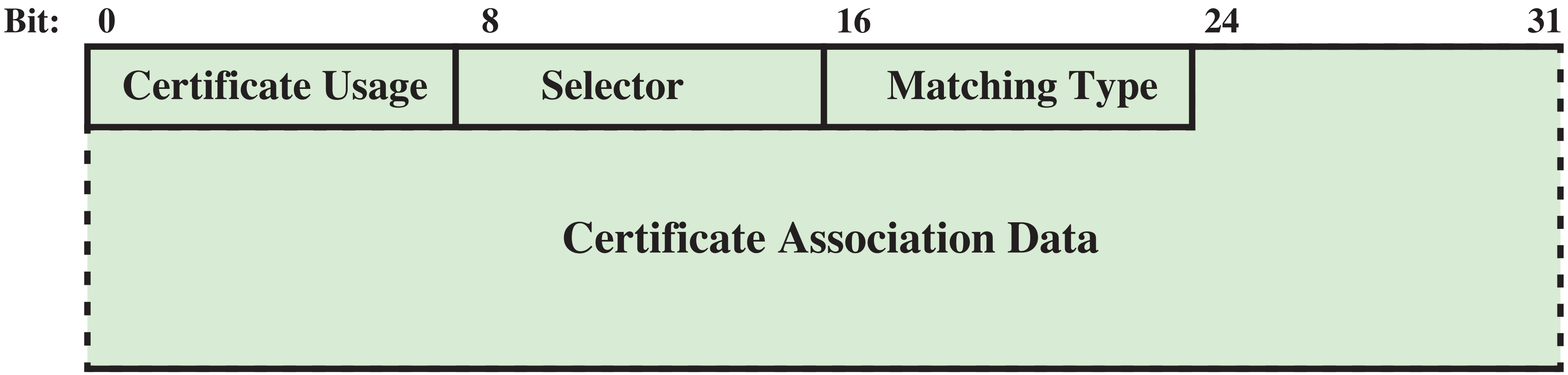
DANE

DNS-Based Authentication of Named Entities

Is a protocol to allow X.509 certificates, commonly used for Transport Layer Security (TLS) to be bound to DNS names using DNSSEC

It is proposed in RFC 6698 as a way to authenticate TLS client and server entities **without** a certificate authority (CA)

The purpose of DANE is to replace reliance on the security of the CA system with reliance on the security provided by DNSSEC



Bit: 0 8 16 24 31

Certificate Usage

Selector

Matching Type

Certificate Association Data

Figure 19.5 TLSA RR Transmission Format

Sender Policy Framework (SPF)

- SPF is the standardized way for a sending domain to identify and assert the mail senders for a given domain
- RFC 7208 defines the SPF
  - It provides a protocol by which ADMDs can authorize hosts to use their domain names in the “**MAIL FROM**” or “**HELLO**” identities
- SPF works by checking a sender’s IP address against the policy encoded in any SPF record found at the sending domain
  - This means that SPF checks can be applied before the message content is received from the sender

```
S: 220 foo.com Simple Mail Transfer Service Ready
C: HELO mta.example.net
S: 250 OK
C: MAIL FROM:<alice@example.org>
S: 250 OK
C: RCPT TO:<Jones@foo.com>
S: 250 OK
C: DATA
S: 354 Start mail input; end with <CRLF>.<CRLF>
C: To: bob@foo.com
C: From: alice.sender@example.net
C: Date: Today
C: Subject: Meeting Today
. . .
```

Figure 19.6 Example in which SMTP Envelope Header Does Not Match Message Header

Tag (Name)	Description
v= (Version)	Version field that must be present as the first element. By default the value is always DMARC1.
p= (Policy)	Mandatory policy field. May take values none or quarantine or reject. This allows for a gradually tightening policy where the sender domain recommends no specific action on mail that fails DMARC checks (p=none), through treating failed mail as suspicious (p=quarantine), to rejecting all failed mail (p=reject), preferably at the SMTP transaction stage.
aspf= (SPF Policy)	Values are r (default) for relaxed and s for strict SPF domain enforcement. Strict alignment requires an exact match between the From address domain and the (passing) SPF check must exactly match the MailFrom address (HELO address). Relaxed requires that only the From and MailFrom address domains be in alignment. For example, the MailFrom address domain smtp.example.org and the From address announce@example.org are in alignment, but not a strict match.
adkim = (DKIM Policy)	Optional. Values are r (default) for relaxed and s for strict DKIM domain enforcement. Strict alignment requires an exact match between the From domain in the message header and the DKIM domain presented in the d= DKIM tag. Relaxed requires only that the domain part is in alignment (as in aspf).
fo= (Failure reporting options)	Optional. Ignore if a ruf argument is not also present. Value 0 indicates the receiver should generate a DMARC failure report if all underlying mechanisms fail to produce an aligned pass result. Value 1 means generate a DMARC failure report if any underlying mechanism produces something other than an aligned pass result. Other possible values are d (generate a DKIM failure report if a signature failed evaluation), and s (generate an SPF failure report if the message failed SPF evaluation). These values are not exclusive and may be combined.
ruf=	Optional, but requires the fo argument to be present. Lists a series of URIs (currently just mailto:<emailaddress>) that list where to send forensic feedback reports. This is for reports on message specific failures.
rua=	Optional list of URIs (like in ruf=, using the mailto: URI) listing where to send aggregate feedback back to the sender. These reports are sent based on the interval requested using the ri= option, with a default of 86400 seconds if not listed.
ri= (Reporting interval)	Optional with the default value of 86400 seconds. The value listed is the reporting interval desired by the sender.
pct= (Percent)	Optional with the default value of 100. Expresses the percentage of a sender’s mail that should be subject to the given DMARC policy. This allows senders to ramp up their policy enforcement gradually and prevent having to commit to a rigorous policy before getting feedback on their existing policy.
sp= (Receiver policy)	Optional with a default value of none. Other values include the same range of values as the p= argument. This is the policy to be applied to mail from all identified subdomains of the given DMARC RR.

Table 19.7

DMARC Tag and Value Descriptions

(Table is on page 607 in the textbook)

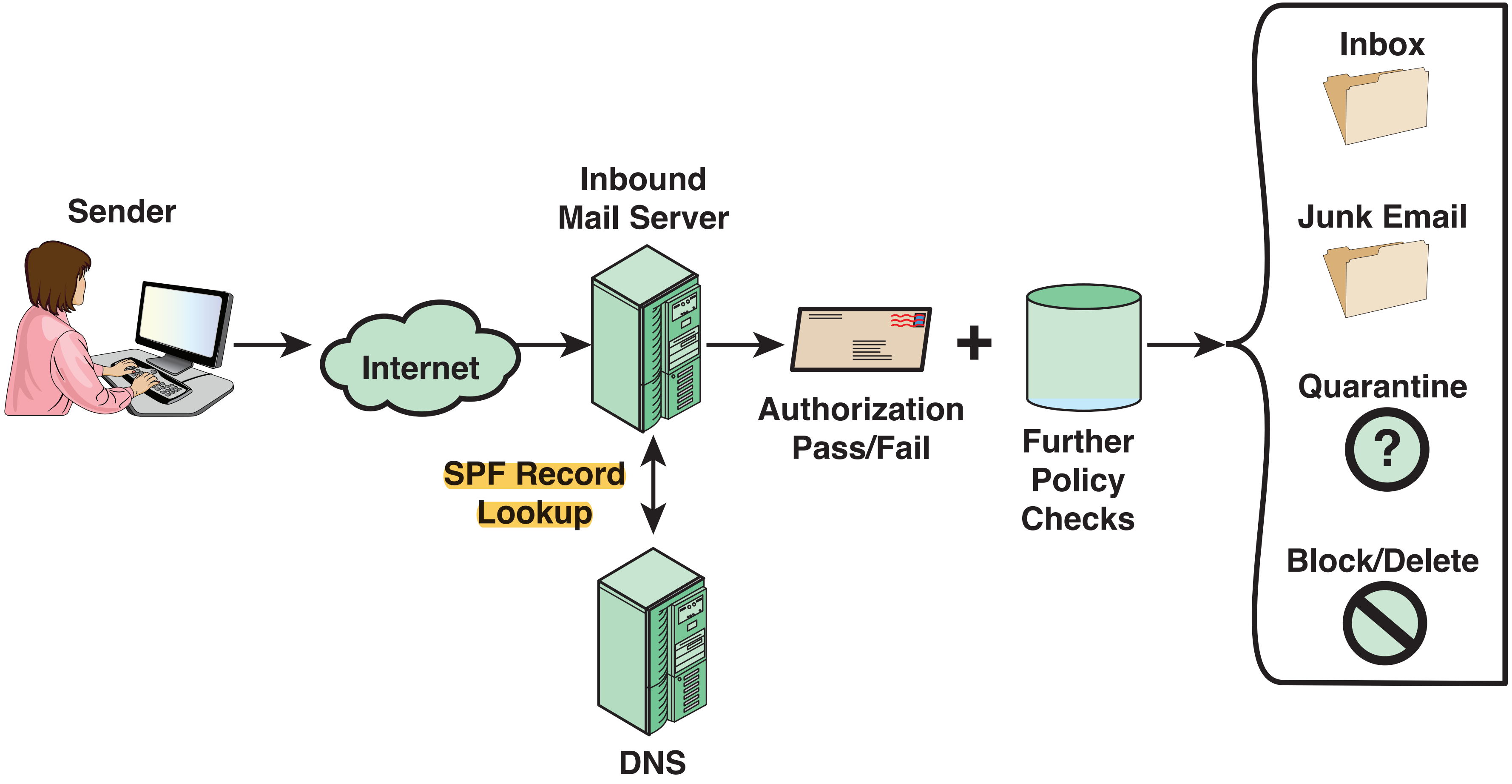


Figure 19.7 Sender Policy Framework Operation

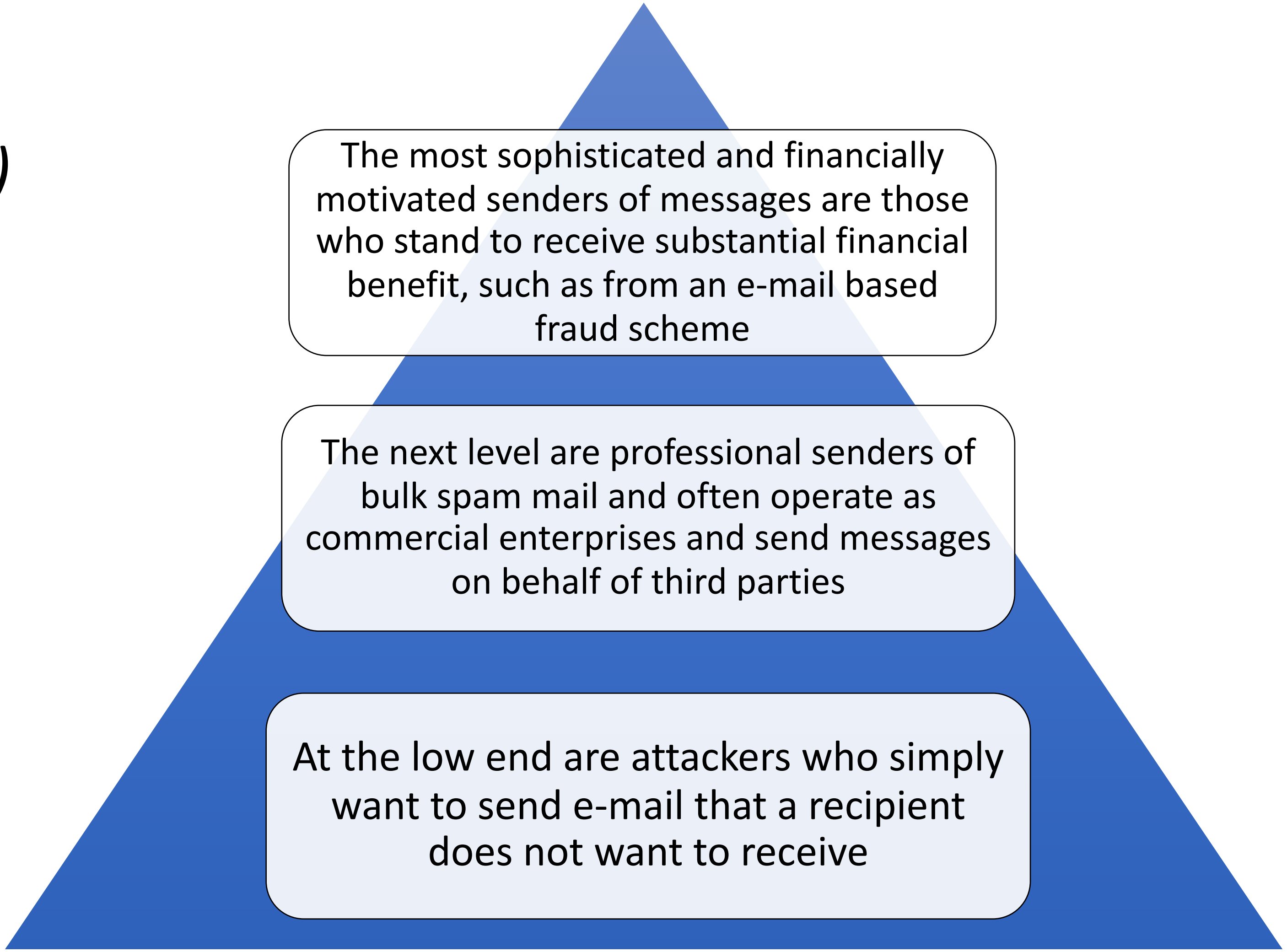
DomainKeys Identified Mail (DKIM)

- A specification for cryptographically signing e-mail messages, permitting a signing domain to claim responsibility for a message in the mail stream
- Message recipients can verify the signature by querying the signer's domain directly to retrieve the appropriate public key and can thereby confirm that the message was attested to by a party in possession of the private key for the signing domain
- Proposed Internet Standard RFC 6376
- Has been widely adopted by a range of e-mail providers and Internet Service Providers (ISPs)

E-mail Threats

- RFC 4686 *(Analysis of Threats Motivating DomainKeys Identified Mail)*
  - Describes the threats being addressed by DKIM in terms of the characteristics, capabilities, and location of potential attackers

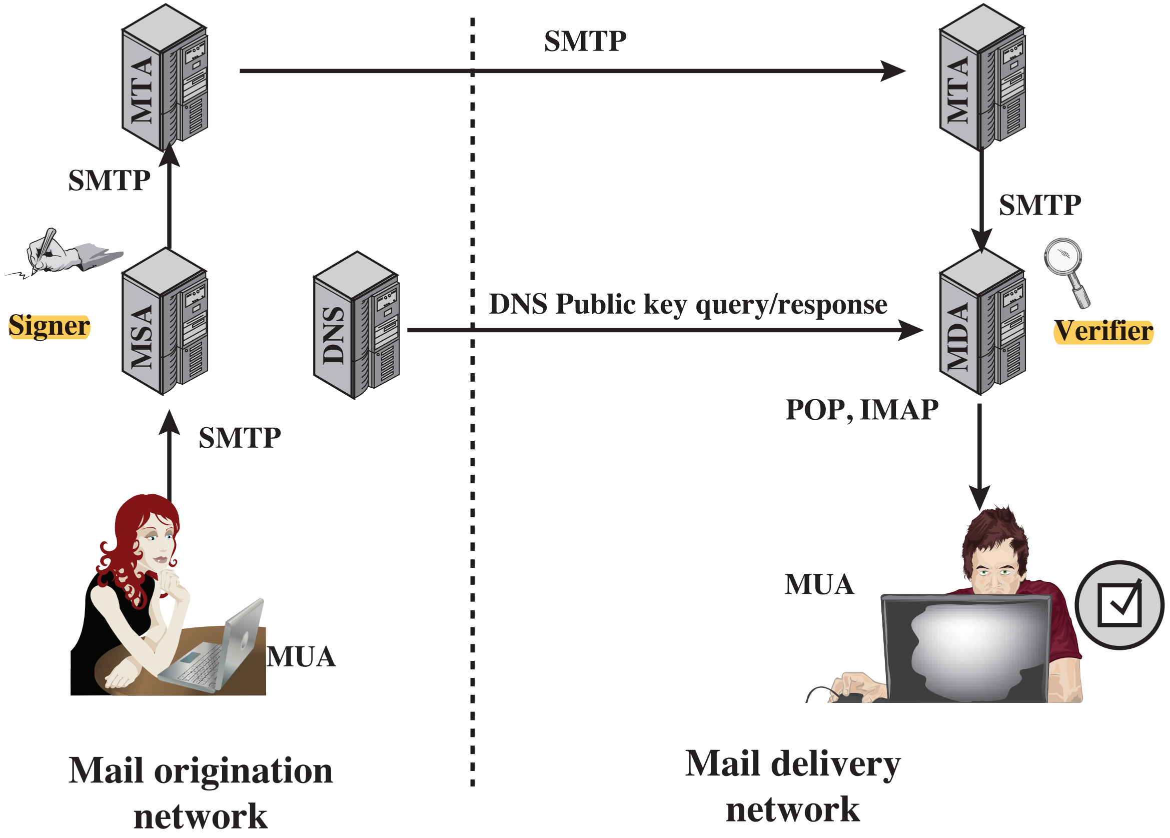
- Characterized on three levels of threat:



The most sophisticated and financially motivated senders of messages are those who stand to receive substantial financial benefit, such as from an e-mail based fraud scheme

The next level are professional senders of bulk spam mail and often operate as commercial enterprises and send messages on behalf of third parties

At the low end are attackers who simply want to send e-mail that a recipient does not want to receive



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

Figure 19.8 Simple Example of DKIM Deployment

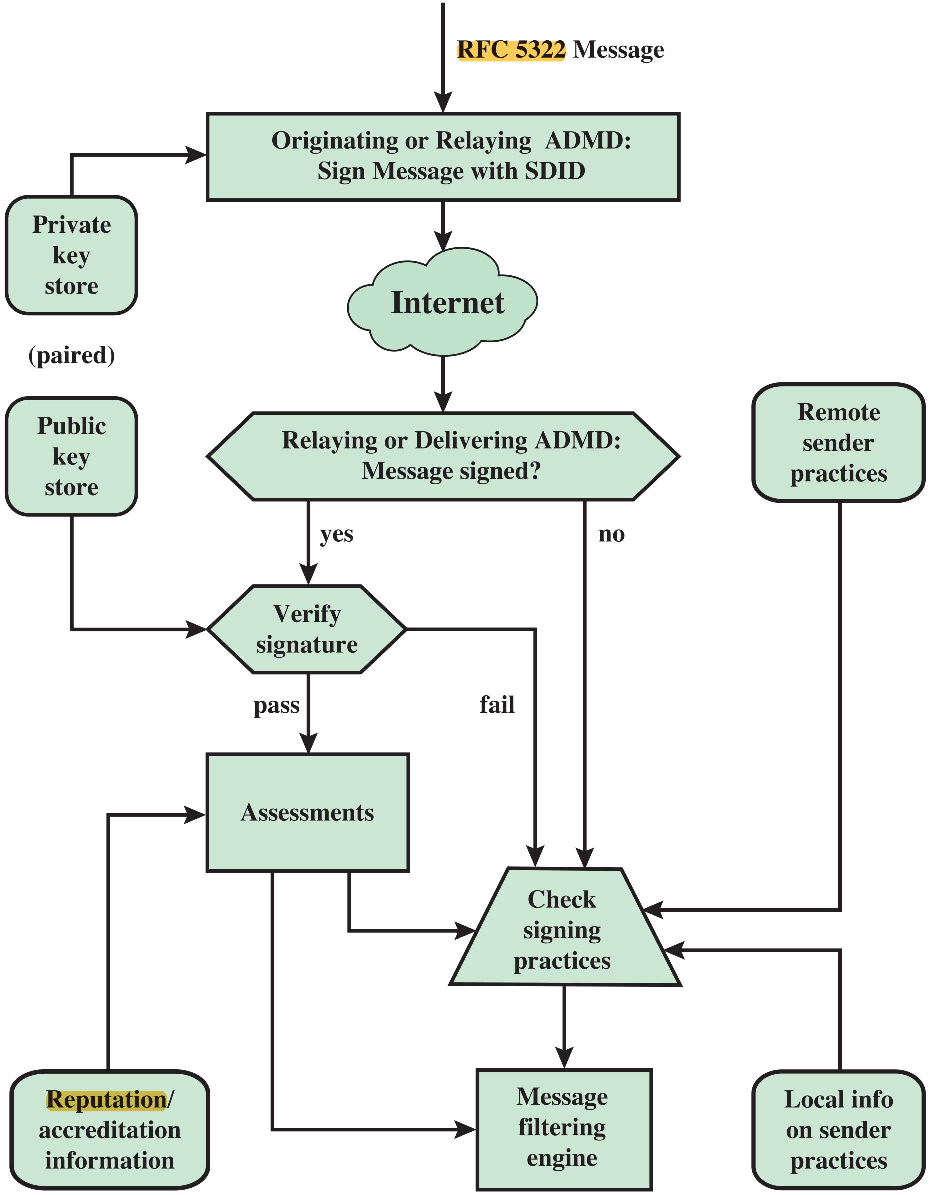


Figure 19.9 DKIM Functional Flow

DMARC

- Domain-Based Message Authentication, Reporting, and Conformance
- Allows email senders to specify policy on how their mail should be handled, the types of reports that receivers can send back, and the frequency those reports should be sent
- It is defined in RFC 7489
  - *Domain-based Message Authentication, Reporting, and Conformance,* March 2015

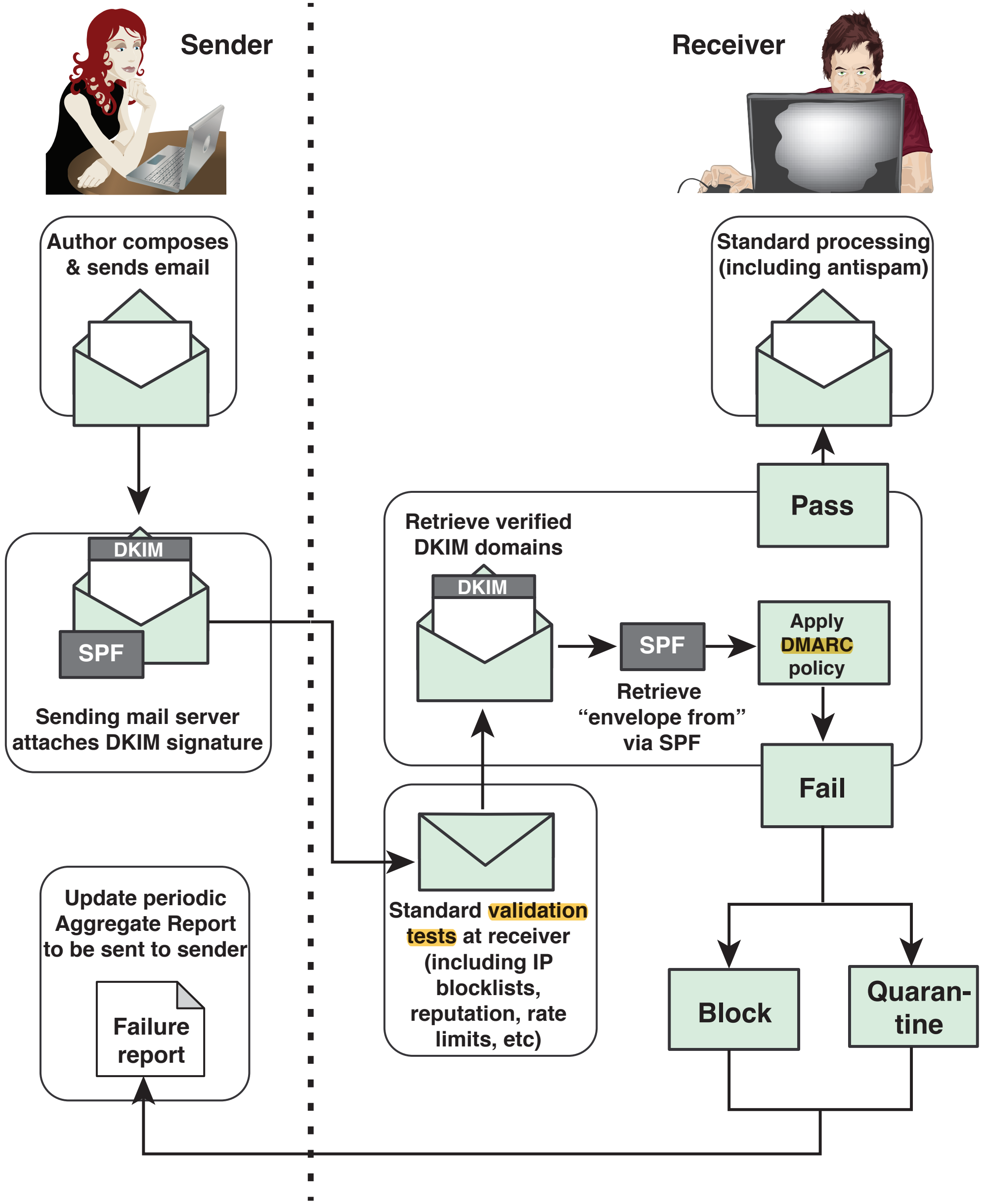

Sender
Receiver
Author composes & sends email
DKIM
SPF
Sending mail server attaches DKIM signature
Update periodic Aggregate Report to be sent to sender
Failure report
Standard validation tests at receiver (including IP blocklists, reputation, rate limits, etc)
Retrieve verified DKIM domains
DKIM
SPF
Retrieve "envelope from" via SPF
Apply DMARC policy
Pass
Fail
Block
Quarantine
Standard processing (including antispam)

Figure 19.10 DMARC Functional Flow

Tag (Name)	Description
v= (Version)	Version field that must be present as the first element. By default the value is always DMARC1.
p= (Policy)	Mandatory policy field. May take values none or quarantine or reject. This allows for a gradually tightening policy where the sender domain recommends no specific action on mail that fails DMARC checks (p=none), through treating failed mail as suspicious (p=quarantine), to rejecting all failed mail (p=reject), preferably at the SMTP transaction stage.
aspf= (SPF Policy)	Values are r (default) for relaxed and s for strict SPF domain enforcement. Strict alignment requires an exact match between the From address domain and the (passing) SPF check must exactly match the MailFrom address (HELO address). Relaxed requires that only the From and MailFrom address domains be in alignment. For example, the MailFrom address domain smtp.example.org and the From address announce@example.org are in alignment, but not a strict match.
adkim = (DKIM Policy)	Optional. Values are r (default) for relaxed and s for strict DKIM domain enforcement. Strict alignment requires an exact match between the From domain in the message header and the DKIM domain presented in the d= DKIM tag. Relaxed requires only that the domain part is in alignment (as in aspf).
fo= (Failure reporting options)	Optional. Ignore if a ruf argument is not also present. Value 0 indicates the receiver should generate a DMARC failure report if all underlying mechanisms fail to produce an aligned pass result. Value 1 means generate a DMARC failure report if any underlying mechanism produces something other than an aligned pass result. Other possible values are d (generate a DKIM failure report if a signature failed evaluation), and s (generate an SPF failure report if the message failed SPF evaluation). These values are not exclusive and may be combined.

Table 19.7

DMARC Tag and Value Descriptions

(Table is on page 615 in the textbook)

Tag (Name)	Description
ruf=	Optional, but requires the fo argument to be present. Lists a series of URIs (currently just mailto:<emailaddress>) that list where to send forensic feedback reports. This is for reports on message specific failures.
rua=	Optional list of URIs (like in ruf=, using the mailto: URI) listing where to send aggregate feedback back to the sender. These reports are sent based on the interval requested using the ri= option, with a default of 86400 seconds if not listed.
ri= (Reporting interval)	Optional with the default value of 86400 seconds. The value listed is the reporting interval desired by the sender.
pct= (Percent)	Optional with the default value of 100. Expresses the percentage of a sender’s mail that should be subject to the given DMARC policy. This allows senders to ramp up their policy enforcement gradually and prevent having to commit to a rigorous policy before getting feedback on their existing policy.
sp= (Receiver policy)	Optional with a default value of none. Other values include the same range of values as the p= argument. This is the policy to be applied to mail from all identified subdomains of the given DMARC RR.

Table 19.7

DMARC Tag and Value Descriptions

(Table is on page 615 in the textbook)

Summary

- Summarize the key functional components of the Internet mail architecture
- Explain the basic functionality of SMTP, POP3, and IMAP
- Explain the need for MIME as an enhancement to ordinary email
- Describe the key elements of MIME
- Understand the functionality of S/MIME and these security threats it addresses

- Understand the basic mechanisms of STARTTLS and its role in email security
- Understand the basic mechanisms of DANE and its role in email security
- Understand the basic mechanisms of SPF and its role in email security
- Understand the basic mechanisms of DKIM and its role in email security
- Understand the basic mechanisms of DMARC and its role in email security