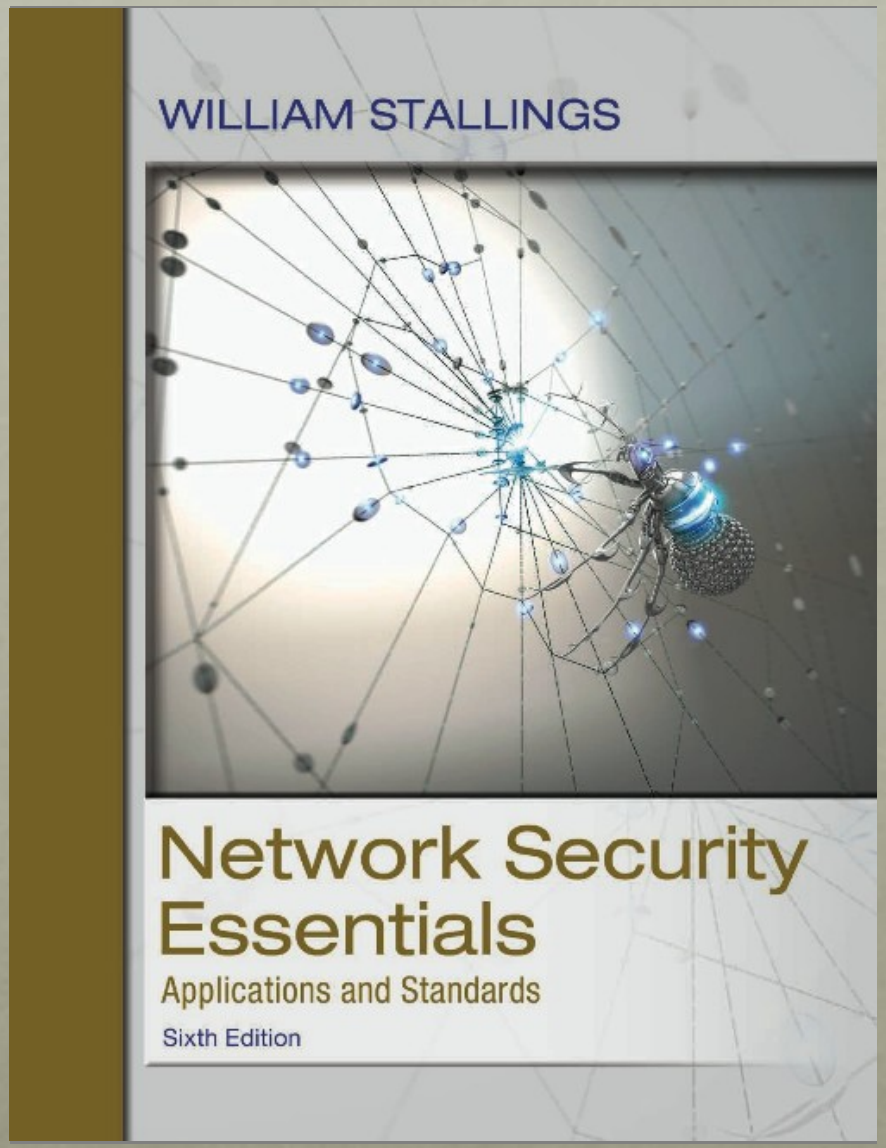
Network Security Essentials

Sixth Edition

by William Stallings



Chapter 8

DNS
Electronic Mail Security

Internet mail architecture

- Defined in RFC 5598 (*Internet Mail Architecture*, July 2009)
- E-mail components:
  - At its most fundamental level, the internet mail architecture consists of a user world in the form of Message User Agents (MUA)
  - And the transfer world, in the form of the Message Handling Service (MHS), which is composed of Message Transfer Agents (MTA)
    - The MHS accepts a message from one user and delivers it to one or more other users, creating a virtual MUA-to-MUA exchange environment
    - This architecture involves three types of interoperability

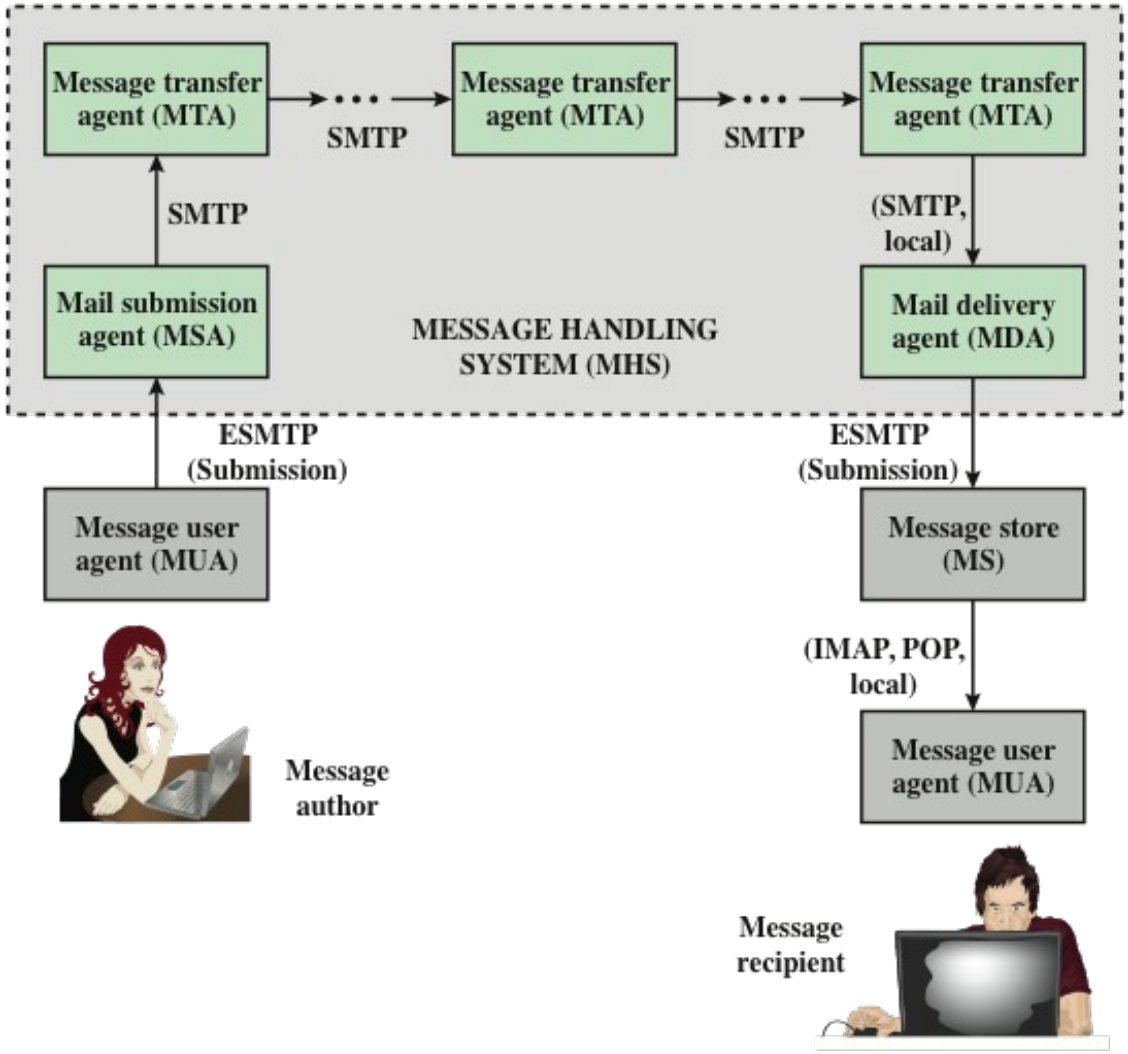


Figure 8.1 Function Modules and Standardized Protocols Used Between Them in the Internet Mail Architecture

E-Mail components

- Administrative management domain (ADMD)
  - Internet e-mail provider
  - Examples include a department that operates a local mail relay, an IT department that operates an enterprise mail relay, and an ISP that operates a public shared e-mail service
  - Each ADMD can have different operating policies and trust-based decision making
- Domain name system (DNS)
  - A directory lookup service that provides a mapping between the name of a host on the Internet and its numerical address

E-Mail Protocols SMTP

- Simple Mail Transfer Protocol (SMTP)
  - Encapsulates an e-mail message in an envelope and is used to relay the encapsulated messages from source to destination through multiple MTAs
  - Was originally specified in 1982 as RFC 821
    - Has undergone several revisions, the most current being RFC5321 (October 2008)
  - Is a text-based client-server protocol where the client (e-mail sender) contacts the server (next-hop recipient) and issues a set of commands to tell the server about the message to be sent, then sending the message itself
    - The majority of these commands are ASCII text messages sent by the client and a resulting return code returned by the server

```
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 8.2 Example SMTP Transaction Scenario

Mail access protocols

Post Office Protocol (POP3) allows an e-mail client (user agent) to download an e-mail from an e-mail server (MTA).

POP3 user agents connect via TCP to the server (typically port 110). The user agent enters a username and password (either stored internally for convenience or entered each time by the user for stronger security). After authorization, the UA can issue POP3 commands to retrieve and delete mail.

As with POP3, Internet Mail Access Protocol (IMAP) also enables an e-mail client to access mail on an e-mail server. IMAP also uses TCP, with server TCP port 143. IMAP is more complex than POP3. IMAP provides stronger authentication than POP3 and provides other functions not supported by POP3.

E-mail formats

RFC 5322

- Defines a format for text messages that are sent using electronic mail
- Has been the standard for Internet-based text mail messages
- 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

E-mail formats

- Multipurpose Internet Mail Extensions (MIME)
  - Extension to the RFC 5322 framework
  - RFCs 2045 through 2049 define MIME
  - The MIME specification includes the following elements:
    - Five new message header fields are defined, which may be included in an RFC 5322 header, providing information about the body of the message
    - 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

Header fields defined in mime

- MIME-Version
  - 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

Table 8.1

MIME Content Types

(Table can be found on page 245 in the textbook)

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 8.2
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 can be found on page 248 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 8.3 Example MIME Message Structure

(Figure 8.3 can be found on page 249 in the textbook)

Canonical and Native Forms

An important concept in MIME and S/MIME is that of canonical form.

Canonical form is a format, appropriate to the content type, that is standardized for use between systems. This is in contrast to native form, which is a format that may be peculiar to a particular system. RFC 2049 defines these two forms as follows:

■ 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. In essence, the data are created in the native form that corresponds to the type specified by the media type.

■ 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.

E-mail threats

For both organizations and individuals, e-mail is both pervasive and especially vulnerable to a wide range of security threats. In general terms, e-mail security threats can be classified as follows:

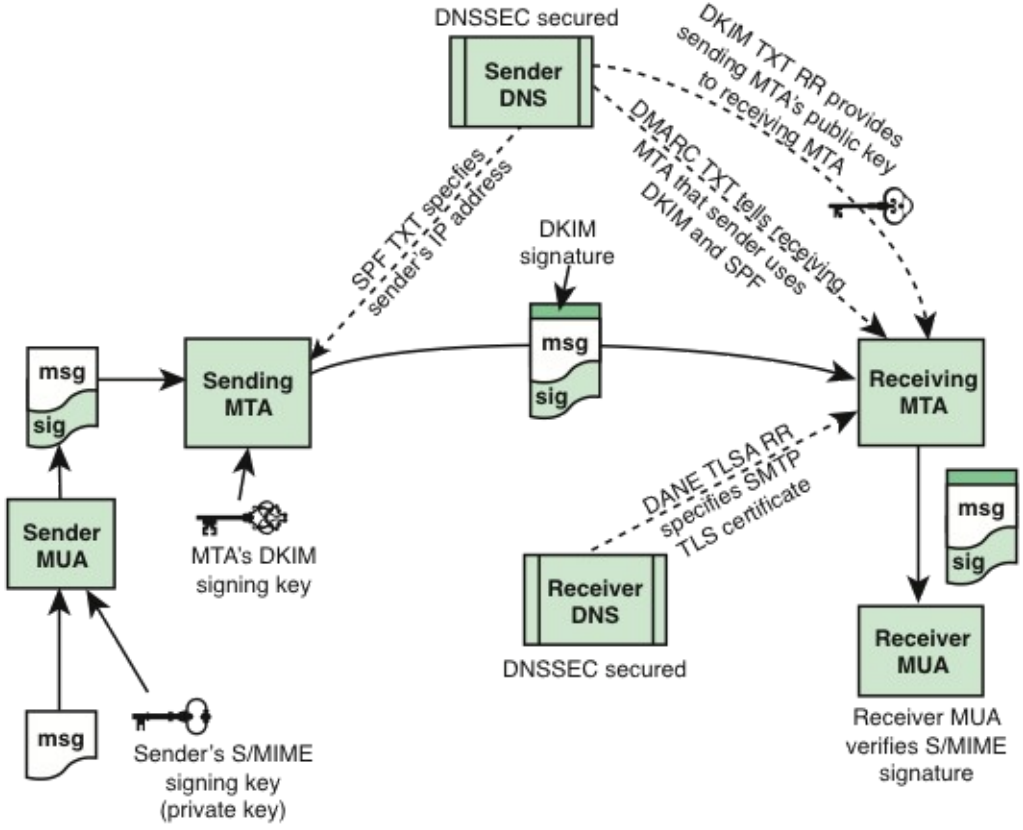
- Authenticity-related threats: Could result in unauthorized access to an enterprise's e-mail system.
- Integrity-related threats: Could result in unauthorized modification of e-mail content.
- Confidentiality-related threats: Could result in unauthorized disclosure of sensitive information.
- Availability-related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

Table 8.3

Email Threats and Mitigations

(Table can be found on page 251 in the textbook)

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.
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.



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

s/mime

Secure/Multipurpose Internet Mail Extension (S/MIME) is a security enhancement to the MIME Internet e-mail format standard based on technology from RSA Data Security. S/MIME is a complex capability that is defined in a number of documents. The most important documents relevant to S/MIME include the following:

■ **RFC 5750, S/MIME Version 3.2 Certificate Handling: Specifies conventions** for X.509 certificate usage by (S/MIME) v3.2.

■ **RFC 5751, S/MIME) Version 3.2 Message Specification: The principal defining** document for S/MIME message creation and processing.

■ **RFC 4134, Examples of S/MIME Messages: Gives examples of message bodies** formatted using S/MIME.

■ **RFC 2634, Enhanced Security Services for S/MIME: Describes four optional** security service extensions for S/MIME.

■ **RFC 5652, Cryptographic Message Syntax (CMS): Describes the Cryptographic** Message Syntax (CMS). This syntax is used to digitally sign, digest, authenticate, or encrypt arbitrary message content.

■ **RFC 3370, CMS Algorithms: Describes the conventions for using several** cryptographic algorithms with the CMS.

■ **RFC 5752, Multiple Signatures in CMS: Describes the use of multiple, parallel** signatures for a message.

■ **RFC 1847, Security Multiparts for MIME—Multipart/Signed and Multipart/** Encrypted: Defines a framework within which security services may be applied to MIME body parts. The use of a digital signature is relevant to S/MIME, as explained subsequently.

Table 8.4
Summary of S/MIME Services

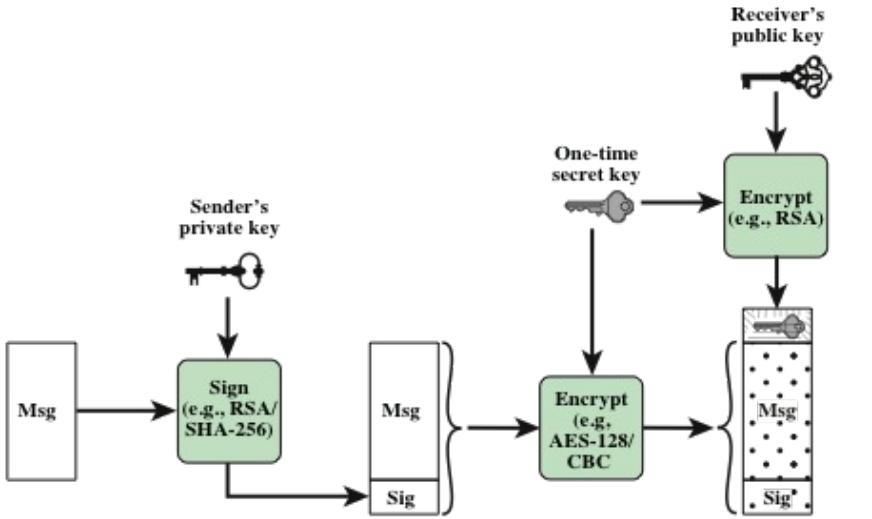
Function	Typical Algorithm	Typical Action
Digital signature	RSA/SHA-256	A hash code of a message is created using SHA-256. This message digest is encrypted using SHA-256 with the sender's private key and included with the message.
Message encryption	AES-128 with CBC	A message is encrypted using AES-128 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.

authentication

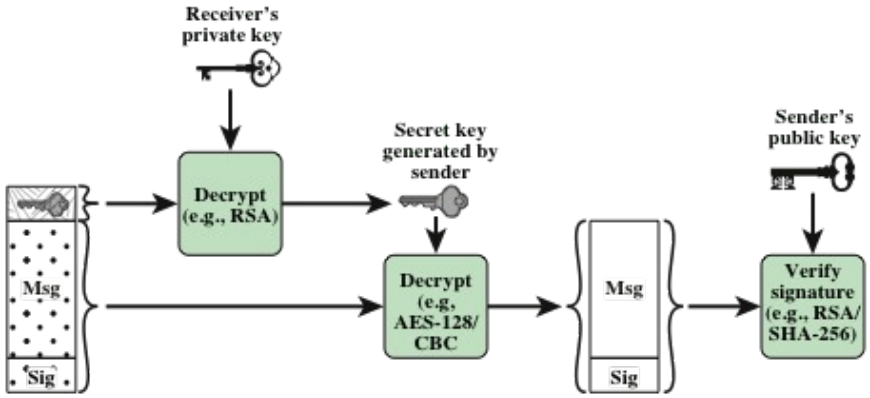
- Authentication is provided by means of a digital signature
  - Most commonly RSA with SHA-256 is used
    - The sender creates a message
    - SHA-256 is used to generate a 256-bit message digest of the message
    - The message digest is encrypted with RSA using the sender's private key, and the result 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

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
- In S/MIME each symmetric key, referred to as a content-encryption key, is used only once
- To protect the key, it is encrypted with the receiver's public key
- Sequence:
  - The sender generates a message and a random 128-bit number to be used as a content-encryption key for this message only
  - The message is encrypted using the content-encryption key
  - The content-encryption key is encrypted with RSA using the recipient's public key and is attached to the message
  - The receiver uses RSA with its private key to decrypt and recover the content-encryption key
  - The content-encryption key is used to decrypt the message



(a) Sender signs, then encrypts message

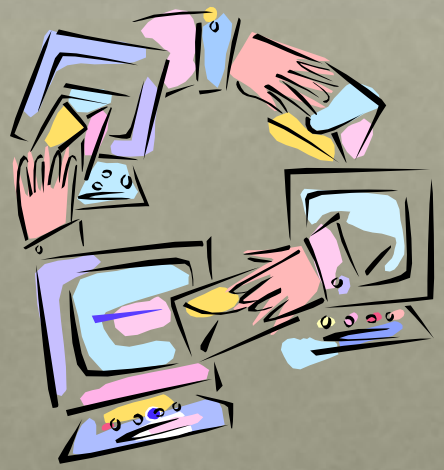


(b) Receiver decrypts message, then verifies sender's signature

Figure 8.5 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
  - A process referred to as 7-bit encoding
- The scheme typically used for this purpose is Base64 conversion
  - Each group of three octets of binary data is mapped into four ASCII characters



compression

- S/MIME also offers the ability to compress a message
  - This has the benefit of saving space both for e-mail transmission and for file storage
- Compression can be applied in any order with respect to the signing and message encryption operations
- RFC 5751 provides the following 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

S/MIME uses the following message content types, which are 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 encryptedcontent
encryption keys for one or more recipients.

■ CompressedData: Used to apply data compression to a message.

The Data content type is also used for a procedure known as clear signing. For 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. Unlike
SignedData, which involves encapsulating the message and signature in a special
format, clear-signed messages can be read and their signatures verified by e-mail
entities that do not implement S/MIME.

Table 8.5
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

s/mime Content types – envelopedData

- The steps for preparing an envelopedData MIME entity are:
  - Generate a pseudorandom session key for a particular symmetric encryption algorithm (RC2/40 or triple DES)
  - 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
  - Encrypt the message content with the session key

s/mime Content types – signedData

The signedData smime-type can be used with one or more signers.
For clarity, we confine our description to the case of a single digital signature. The steps for preparing a signedData MIME entity are as follows.

1. Select a message digest algorithm (SHA or MD5).

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

3. Encrypt the message digest with the signer's private key.

4. 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.

The signedData entity consists of a series of blocks, including a message digest algorithm identifier, the message being signed, and SignerInfo . The signedData entity may also include a set of public-key certificates sufficient to
constitute a chain from a recognized root or top-level certification authority to the signer. This information is then encoded into base64. A sample message (excluding the RFC 5322 headers) is the following.

```
Content-Type: application/pkcs7-mime; smime-type=signeddata;
name=smime.p7m
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename=smime.p7m
567GhIGfHfYT6ghyHhHUujpfyF4f8HHGTrfvhJhjH776tbB9HG4VQbnj7
77n8HHGT9HG4VQpfyF467GhIGfHfYT6rfvbnj756tbBghyHhHUujhJhjH
HUujhJh4VQpfyF467GhIGfHfYGTrfvbnjT6jH7756tbB9H7n8HHGghyHh
6YT64V0GhIGfHfQbnj75
```

To recover the signed message and verify the signature, the recipient first strips off the base64 encoding. Then the signer's public key is used to decrypt the message digest. The recipient independently computes the message digest and compares it to the decrypted message digest to verify the signature.

S/mime certificate processing

- A 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
    - Each key pair MUST be generated from a good source of nondeterministic random input and be protected in a secure fashion
    - 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
      - Such a list could be maintained by the user or by some local administrative entity on behalf of a number of users

Enhanced security services

RFC 2634 defines four enhanced security services for S/MIME:

■ Signed receipts: A signed receipt may be requested in a SignedData object.
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. In essence, the recipient signs the entire
original message plus the original (sender's) signature and appends the new
signature to form a new S/MIME message.

■ Security labels: A security label may be included in the authenticated attributes
of a SignedData object. A security label is a set of security information
regarding the sensitivity of the content that is protected by S/MIME
encapsulation.
The labels may be used for access control, by indicating which users are
permitted access to an object. Other uses include priority (secret, confidential,
restricted, and so on) or role based, describing which kind of people can see
the information (e.g., patient's health-care team, medical billing agents).

■ Secure mailing lists: When a user sends a message to multiple recipients, a

Pretty good privacy (pgp)

- An alternative e-mail security protocol
- Has essentially the same functionality as S/MIME
- Created by Phil Zimmerman and implemented as a product first released in 1991
- It was made available free of charge and became quite popular for personal use
- The initial PGP protocol was proprietary and used some encryption algorithms with intellectual property restriction
- There are two significant differences between S/MIME and OpenPGP:
  - Key certification
  - Key distribution
- SP 800-177 recommends the use of S/MIME rather than PGP because of the greater confidence in the CA system of verifying public keys

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
- Comprised of four elements:
  - Domain name space
    - DNS uses a tree-structured name space to identify resources on the Internet
  - DNS database
    - Conceptually, each node and leaf in the name space tree structure names a set of information that is contained in resource record. The collection of all RRs is organized into a distributed database
  - Name servers
    - These are server programs that hold information about a portion of the domain name tree structure and the associated RRs
  - Resolvers
    - These are programs that extract information from name servers in response to client requests. A typical client request is for an IP address corresponding to a given domain name

Dns database

- DNS is based on a hierarchical database containing resource records (RRs) that include the name, IP address, and other information about hosts
- Key features of the database:
  - Variable-depth hierarchy for names
    - DNS allows essentially unlimited levels and uses the period (.) as the level delimiter in printed names
  - Distributed database
    - The database resides in DNS servers scattered throughout the Internet
  - Distribution controlled by the database
    - The DNS database is divided into thousands of separately managed zones, which are managed by separate administrators

Table 8.6

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.

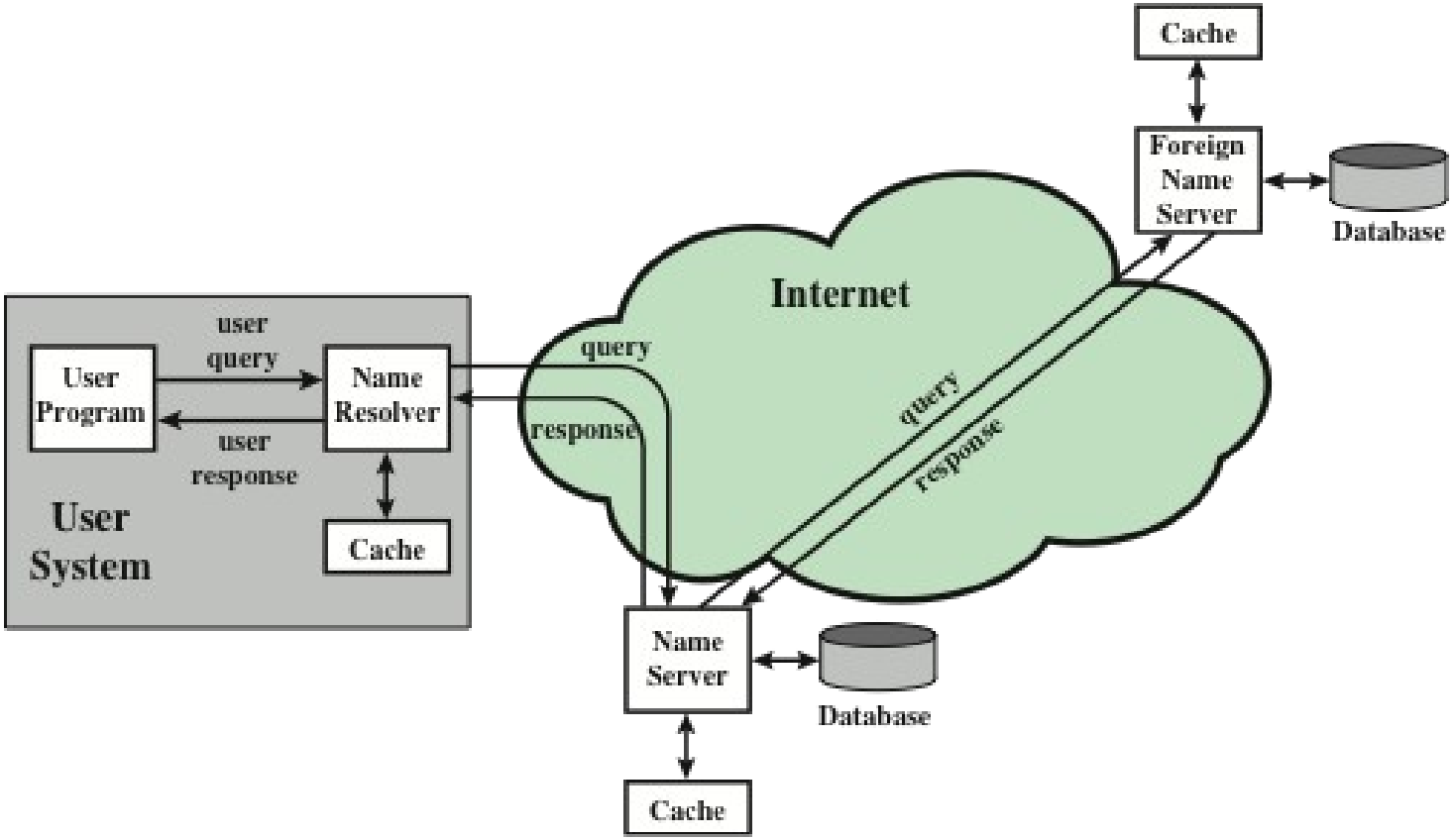


Figure 8.6 DNS Name Resolution

Dns security extensions

- DNSSEC:
  - 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:
  - 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*

Dns-based authentication of named entities (dane)

- DANE 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
- DANE defines a new DNS record type, TLSA, that can be used for a secure method of authenticating SSL/TLS certificates

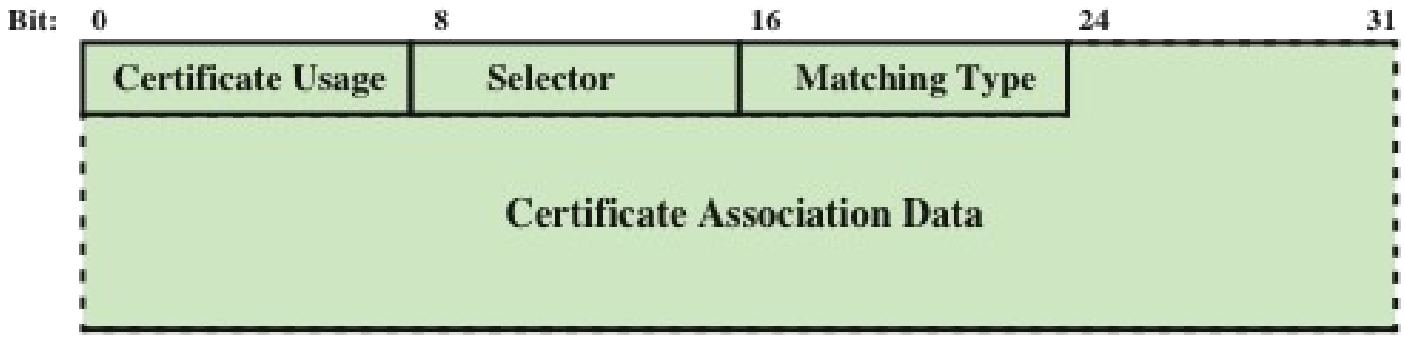


Figure 8.7 TLSA RR Transmission Format

Sender policy framework (spf)

- SPF is the standardized way for a sending domain to identify and assert the mail senders for a given domain
- Addresses the problem of any host being able to use any domain name for each of the various identifiers in the mail header, not just the domain name where the host is located
- Defined in RFC 7208
- SPF works by checking a sender's IP address against the policy encoded in any SPF record found at the sending domain

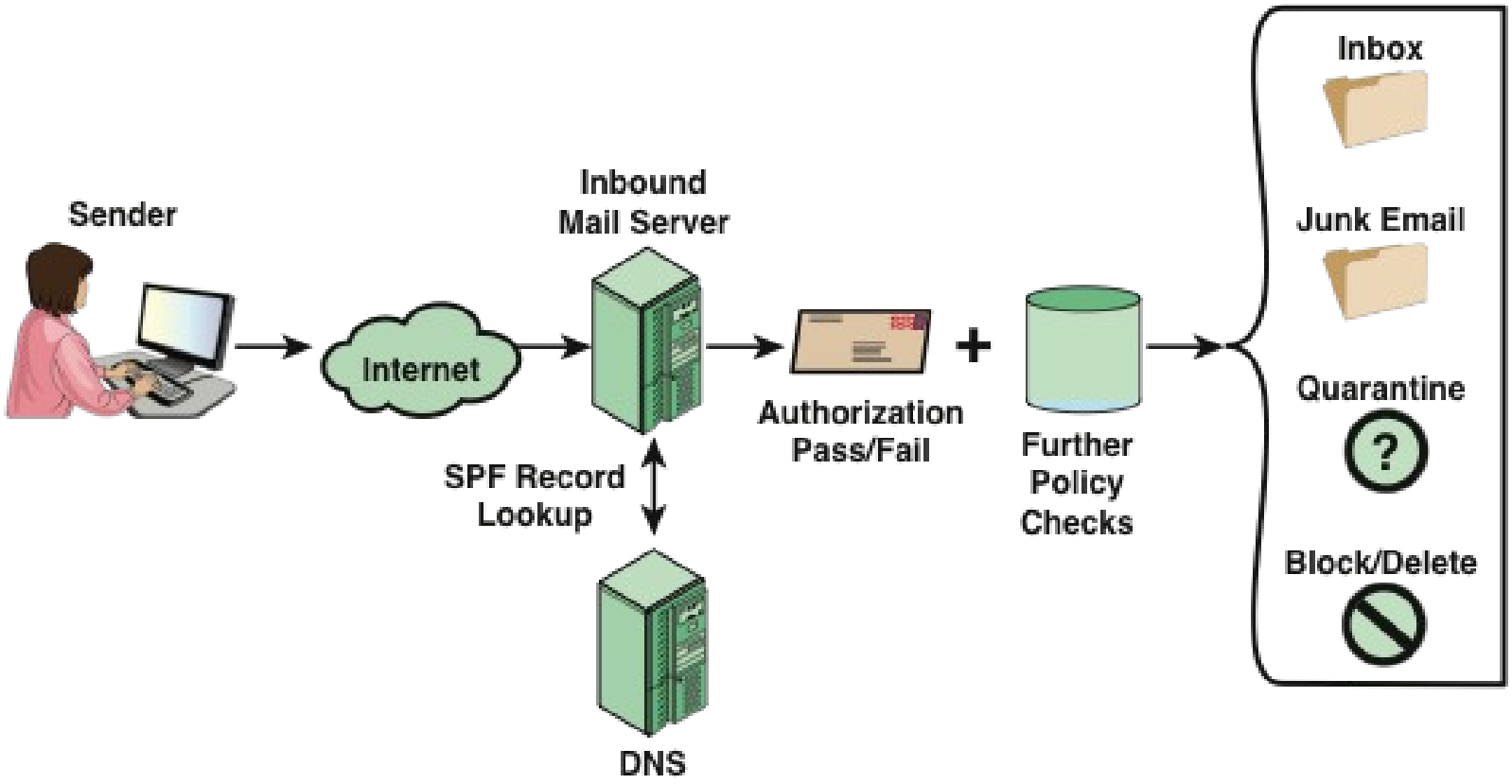


Figure 8.9 Sender Policy Framework Operation

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 characterizes the range of attackers on a spectrum of three levels of threat.

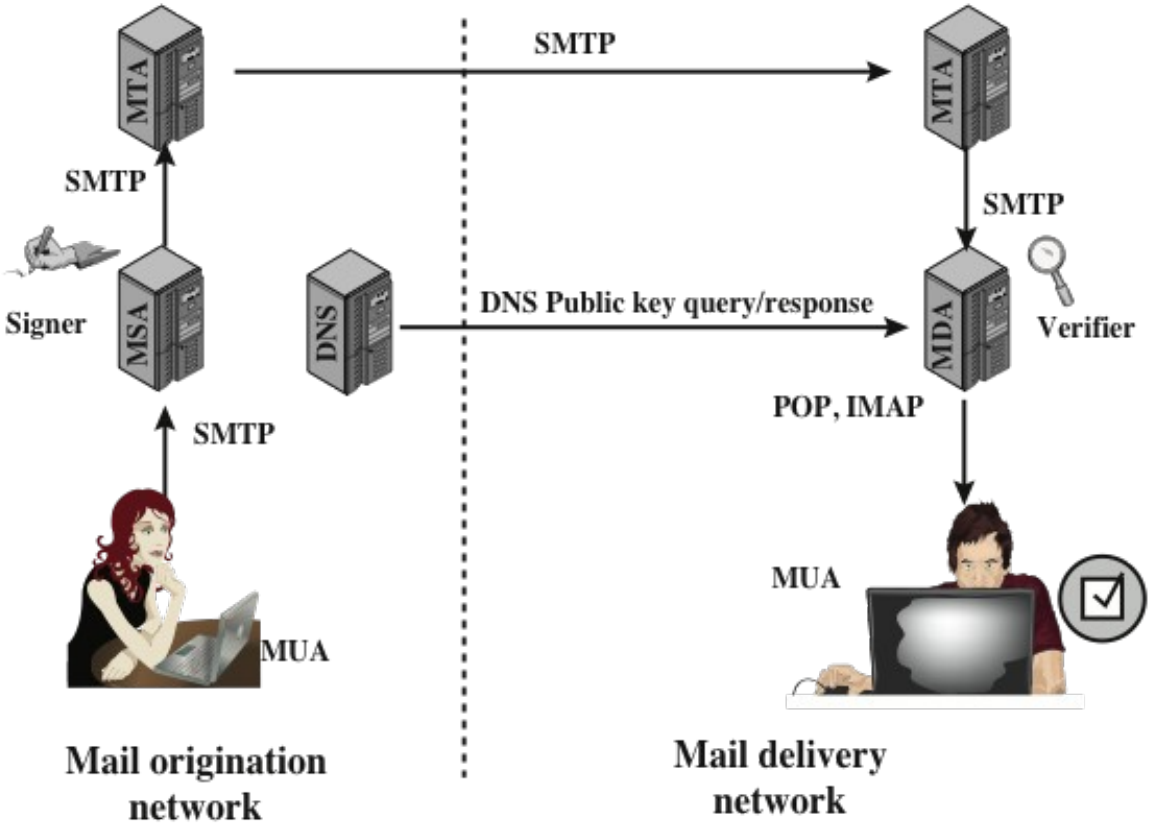
1. At the low end are attackers who simply want to send e-mail that a recipient does not want to receive. The attacker can use one of a number of commercially available tools that allow the sender to falsify the origin address of messages. This makes it difficult for the receiver to filter spam on the basis of originating address or domain.

2. At the next level are professional senders of bulk spam mail. These attackers often operate as commercial enterprises and send messages on behalf of third parties. They employ more comprehensive tools for attack, including Mail Transfer Agents (MTAs) and registered domains and networks of compromised computers (zombies) to send messages and (in some cases) to harvest addresses to which to send.

3. 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. These attackers can be expected to employ all of the above mechanisms and additionally may attack the Internet infrastructure itself, including DNS cache-poisoning attacks and IP routing attacks.

Dkim strategy

- DKIM is designed to provide an e-mail authentication technique that is transparent to the end user
- In essence, a user's e-mail message is signed by a private key of the administrative domain from which the e-mail originates
- The signature covers all of the content of the message and some of the RFC 5322 message headers
- This approach differs from that of S/MIME and PGP, which use the originator's private key to sign the content of the message
- The motivation for DKIM is based on the following reasoning: DKIM is not implemented in client programs (MUAs) and is therefore transparent to the user; the user need not take any action
  - DKIM applies to all mail from cooperating domains
  - DKIM allows good senders to prove that they did send a particular message and to prevent forgers from masquerading as good senders



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

Figure 8.10 Simple Example of DKIM Deployment

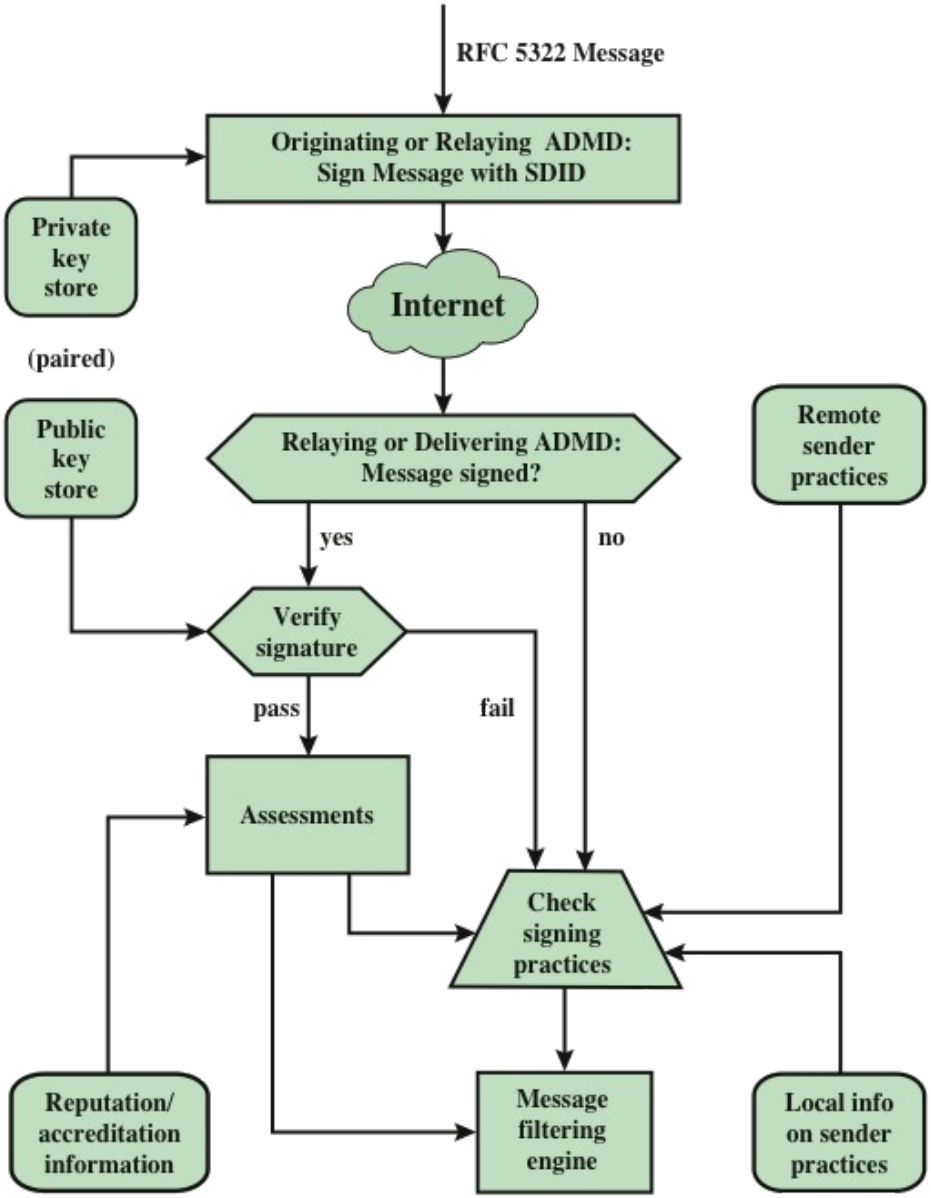


Figure 8.11 DKIM Functional Flow

Domain-based message authentication, reporting, and conformance (dmarc)

- DMARC allows e-mail 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
- Is defined in RFC 7489, *Domain-based Message Authentication, Reporting, and Conformance,* March 2015
- Works with SPF and DKIM
- DMARC standardizes how e-mail receivers perform e-mail authentication using SPF and DKIM mechanisms
- DMARC authentication deals with the From domain in the message header, as defined in RFC 5322
- DMARC requires that From address match (be aligned with) an Authenticated Identifier from DKIM or SPF

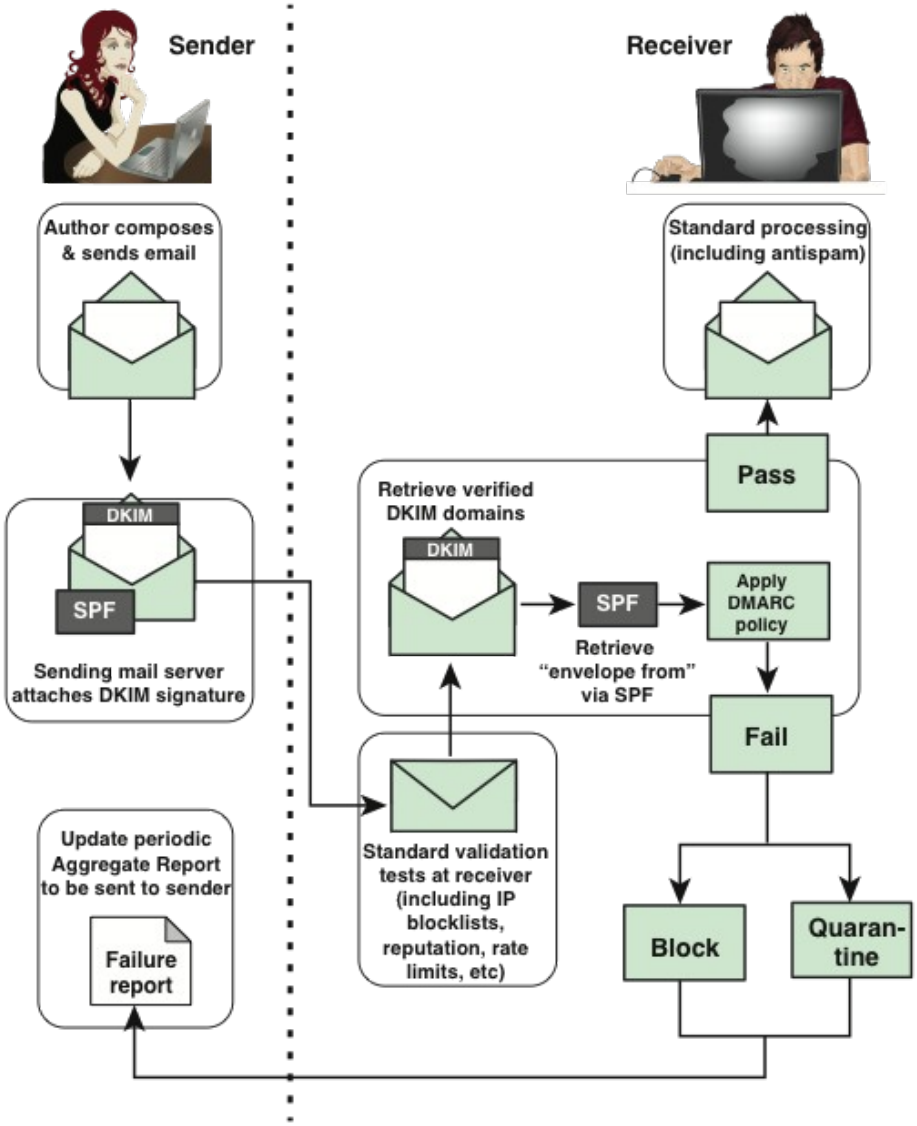


Figure 8.12 DMARC Functional Flow

Table 8.8

DMARC Tag and Value Descriptions

(Table can be found on page 281 in the textbook)

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.

Summary

- Internet mail architecture
- E-mail formats
- E-mail threats and comprehensive e-mail security
- Pretty good privacy
- DomainKeys Identified Mail
- S/MIME
- DNSSEC
- DNS-based authentication of named entities
- Sender policy framework
- Domain-based message authentication, reporting, and conformance