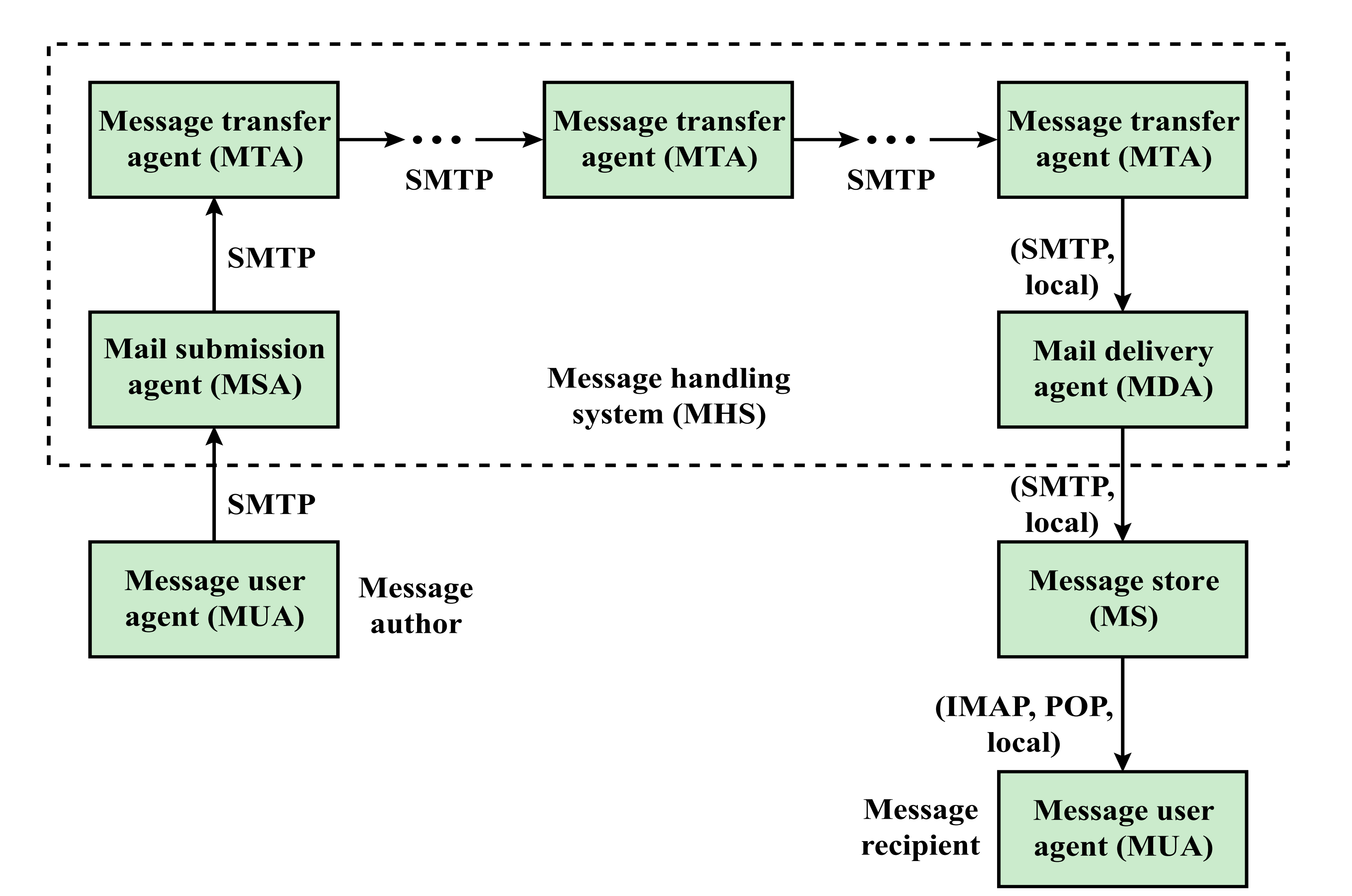
Electronic Mail Security

Ola Flygt
Linnaeus University, Sweden
http://homepage.lnu.se/staff/oflmsi/
Ola.Flygt@lnu.se
+46 470 70 86 49

Outline

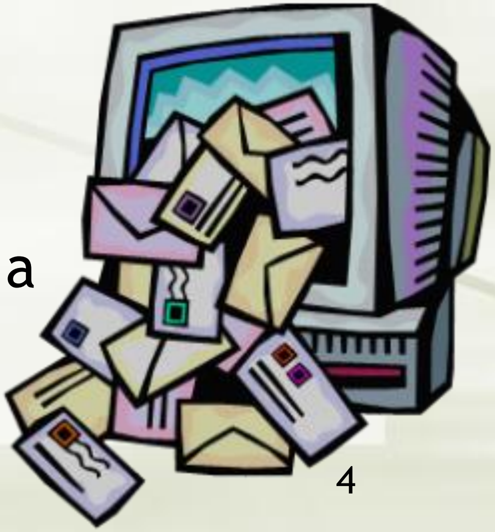
- Email delivery on Internet
- Pretty Good Privacy (PGP)
- S/MIME
- DNS-based authentication of named entities (DANE)
- SMTP Strict Transport Security (SMTP STS)
- DomainKeys Identified Mail (DKIM)
- Sender policy framework (SPF)
- Domain-based Messaging Authentication, Reporting and Conformance (DMARC)

Internet 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)

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



Simple Mail Transfer Protocol (SMTP, RFC 822)

- **SMTP Limitations - Can not transmit, or has a problem with:**
  - executable files, or other binary files (jpeg image)
  - “national language” characters (non-ASCII)
  - messages over a certain size
  - ASCII to EBCDIC translation problems
  - lines longer than a certain length (72 to 254 characters)
- Has undergone several revisions, the most current being RFC5321 (October 2008)

Mail access protocols

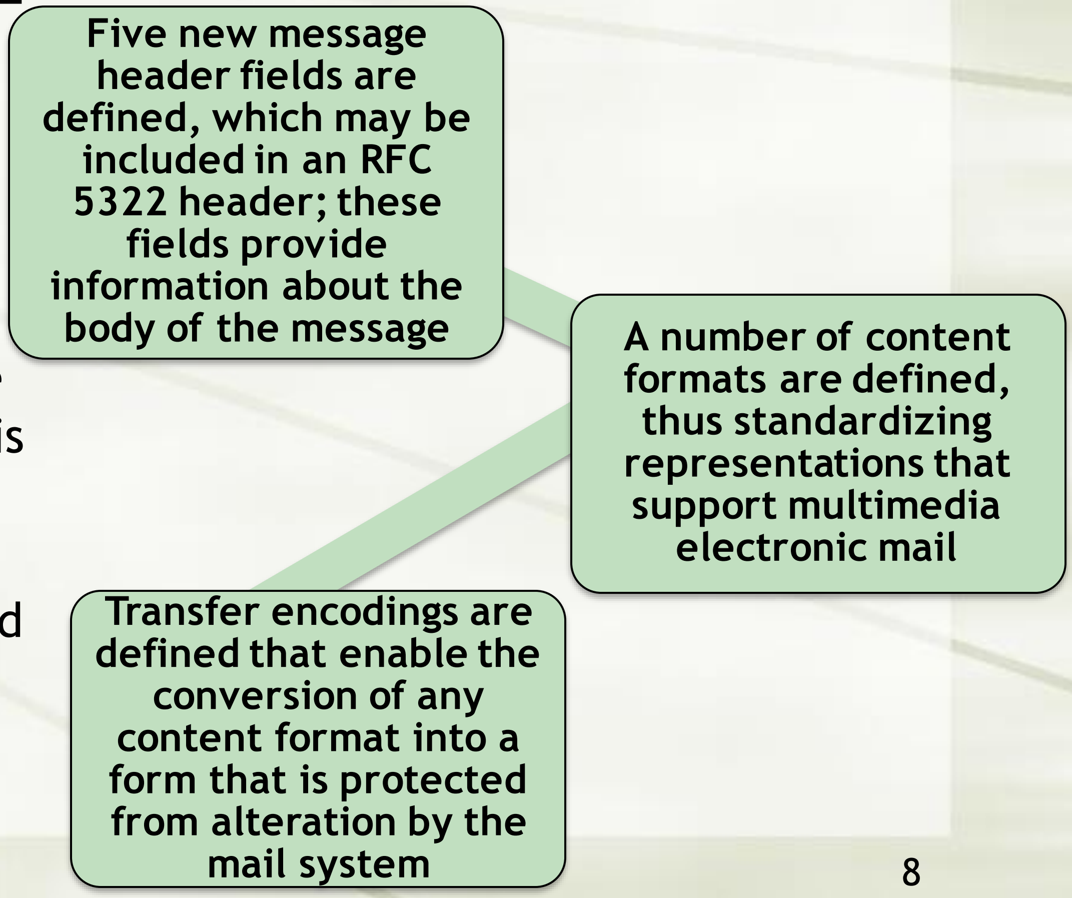
RFC 5322

- 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

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:



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

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

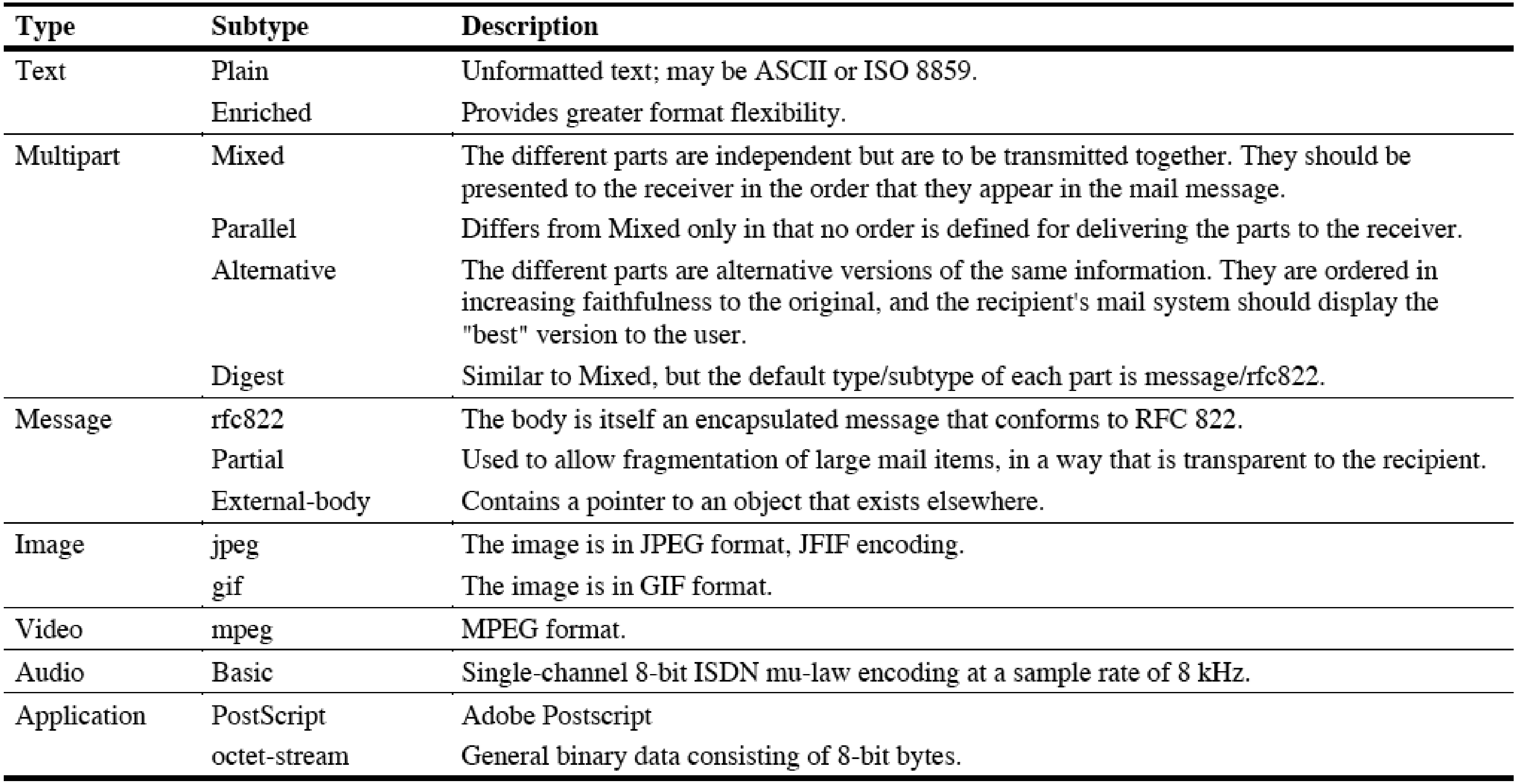
Header fields in MIME

- **MIME-Version:** Must be “1.0” -> RFC 2045, RFC 2046
- **Content-Type:** More types being added by developers (e.g. application/word)
- **Content-Transfer-Encoding:** How message has been encoded (e.g. radix-64)

Optional fields:

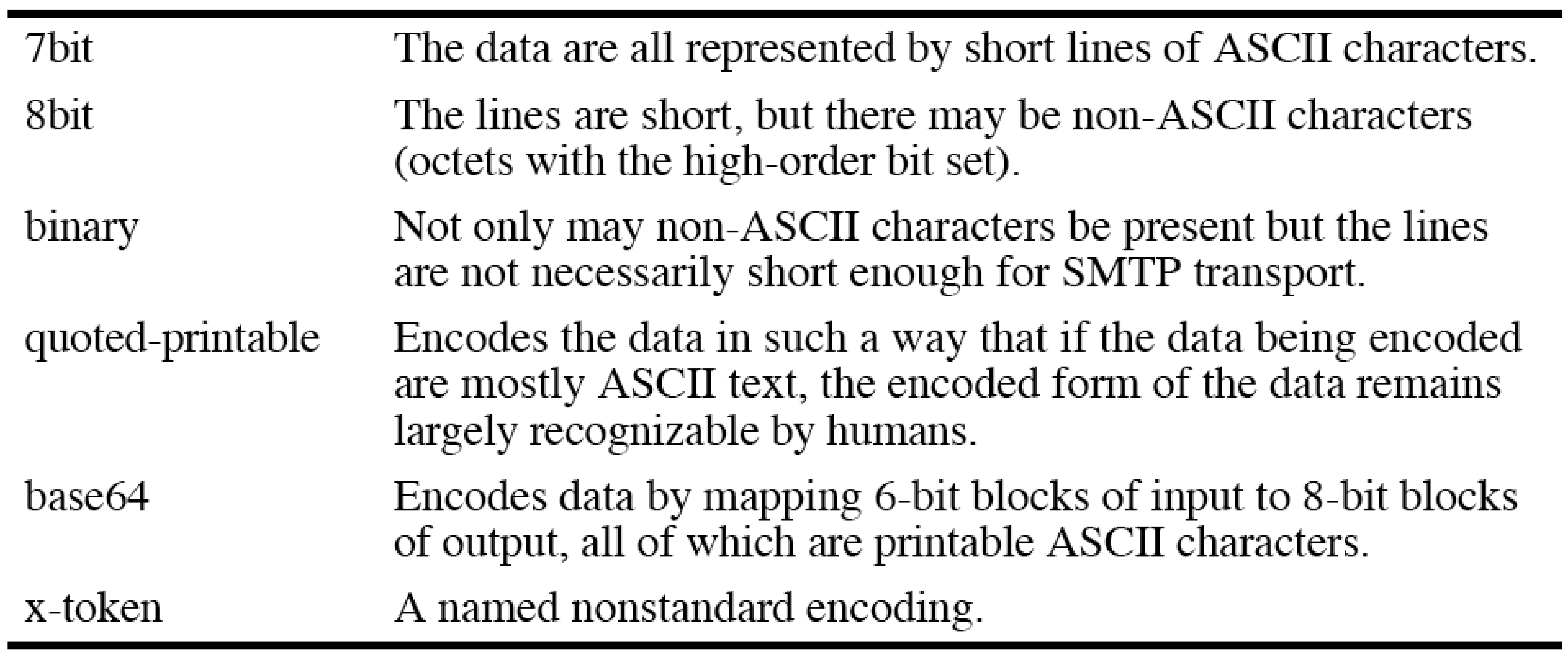
- **Content-ID:** Unique identifying character string.
- **Content Description:** Needed when content is not readable text (e.g. mpeg)

Header fields in MIME,
Content-Type



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.

Header fields in MIME,
Content-Transfer-Encoding



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.

Canonical Form

- An important concept in MIME is that of canonical form. Messages, especially those with attachments, should be in canonical form. The alternative is **Native Form**.
- Conversion into canonical form may include character set conversion, transformation of audio data, compression etc.

MIME Example

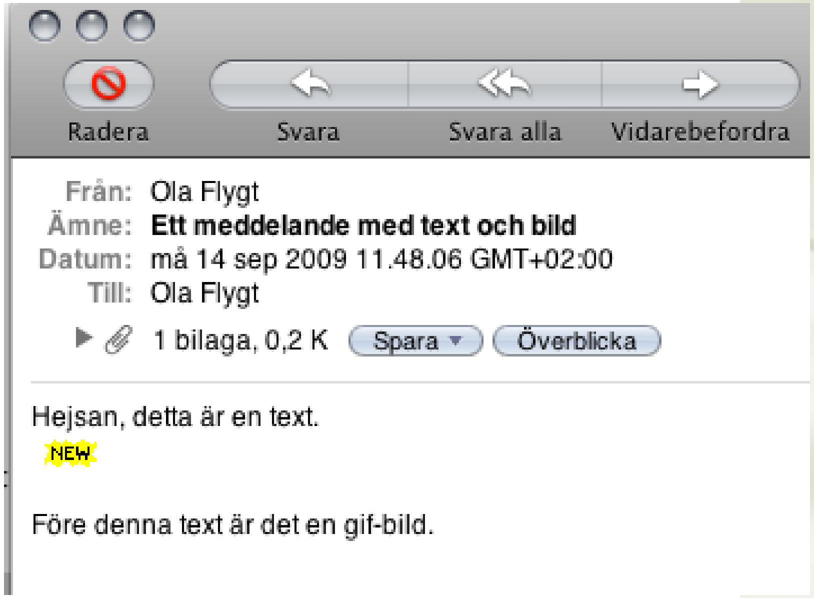
```
Return-Path: <Ola.Flygt@msi.vxu.se>
Received: from webt.msi.vxu.se (localhost [127.0.0.1])
                      by babbage (Cyrus v2.1.16) with LMTP; Wed, 13 Feb 2008 09:22:44 +0100
X-Sieve: CMU Sieve 2.2
Received: from mailinone.vxu.se (mailinone.vxu.se [194.47.65.80])
                      by webt.msi.vxu.se (8.12.10/8.12.10) with ESMTP id m1D8MgvG008161
                      for <Ola.Flygt@msi.vxu.se>; Wed, 13 Feb 2008 09:22:43 +0100 (MET)
Received: from [194.47.95.160] by mailinone.vxu.se
 (Sun Java System Messaging Server 6.2-8.01 (built Nov 27 2006))
 with ESMTPSA id <0JW600AJC4LTR420@mailinone.vxu.se> for
 Ola.Flygt@tcp_msi-daemon (ORCPT Ola.Flygt@msi.vxu.se); Wed,
 13 Feb 2008 09:22:41 +0100 (MET)
Date: Wed, 13 Feb 2008 09:22:39 +0100
From: Ola Flygt <Ola.Flygt@msi.vxu.se>
Subject: A simple MIME example
To: Ola Flygt <Ola.Flygt@msi.vxu.se>
Message-id: <B3AE89DB-D637-43A3-8D2A-55C05311E661@msi.vxu.se>
MIME-version: 1.0 (Apple Message framework v753)
X-Mailer: Apple Mail (2.753)
Content-type:  text/plain; charset=US-ASCII; format=flowed
Content-transfer-encoding: 7bit
X-Spam-Debug:  msi.vxu.se 0
X-Spam-Report:  msi.vxu.se 0 ()
X-Spam-Score: 0
X-Spam-Flag: NO
X-Scanned-By: MIMEDefang 2.49 on 194.47.94.71

<x-flowed>This is a simple text message.

</x-flowed>
```

MIME Example 2

```
Return-Path: <Ola.Flygt@msi.vxu.se>
Received: from webt.msi.vxu.se (localhost [127.0.0.1])
                    by babbage (Cyrus v2.1.16) with LMTP; Wed, 13 Feb 2008 09:24:08 +0100
X-Sieve: CMU Sieve 2.2
Received: from mailinone.vxu.se (mailinone.vxu.se [194.47.65.80])
                    by webt.msi.vxu.se (8.12.10/8.12.10) with ESMTP id m1D8O7vG009405
                    for <Ola.Flygt@msi.vxu.se>; Wed, 13 Feb 2008 09:24:07 +0100 (MET)
Received: from [194.47.95.160] by mailinone.vxu.se
 (Sun Java System Messaging Server 6.2-8.01 (built Nov 27 2006))
 with ESMTPSA id <0JW600AIM4O7R320@mailinone.vxu.se> for
 Ola.Flygt@tcp_msi-daemon (ORCPT Ola.Flygt@msi.vxu.se); Wed,
 13 Feb 2008 09:24:07 +0100 (MET)
Date: Wed, 13 Feb 2008 09:24:05 +0100
From: Ola Flygt <Ola.Flygt@msi.vxu.se>
Subject: A HTML example
To: Ola Flygt <Ola.Flygt@msi.vxu.se>
Message-id: <64F18993-F609-4694-A85A-8BD5A3D007EB@msi.vxu.se>
MIME-version: 1.0
X-Mailer: Apple Mail (2.753)
Content-type: multipart/alternative; boundary=Apple-Mail-46--901936981
X-Spam-Debug: msi.vxu.se 0
X-Spam-Report: msi.vxu.se 0 ()
X-Spam-Score: 0
X-Spam-Flag: NO
X-Scanned-By: MIMEDefang 2.49 on 194.47.94.71

<x-html><!x-stuff-for-pete base="" src="" id="0" charset=""><html><body style="word-wrap: break-word; -webkit-nbsp-mode: space; -webkit-line-break:
after-white-space; "><font class="Apple-style-span" face="Verdana">This is a mail using </font><font class="Apple-style-span"
face="Verdana"><b>HTML</b></font><font class="Apple-style-span" face="Verdana"> encoding of the text.</font>
</body></html>
</x-html>
```

MIME Example 3

```
From: Ola Flygt <ola.flygt@vxu.se>
To: Ola Flygt <ola.flygt@vxu.se>
Content-type: multipart/mixed; boundary=Apple-Mail-5--263420395
MIME-version: 1.0 (Apple Message framework v936)
Subject: Ett meddelande med text och bild
Date: Mon, 14 Sep 2009 11:48:06 +0200

--Apple-Mail-5–263420395
Content-Type: text/plain;
              charset=ISO-8859-1;
              format=flowed
Content-Transfer-Encoding: quoted-printable

Hejsan, detta =E4r en text.

--Apple-Mail-5–263420395
Content-Disposition: inline;
              filename=new01.gif
Content-Type: image/gif;
              x-unix-mode=0644;
              name="new01.gif"
Content-Transfer-Encoding: base64

R0lGODlhIwAUALMIAAAAAIAAAACAAICAAAAAgIAAgACAgICAgMDAwP8AAAD/AP//AAAA//8A/wD/
/////yH5BAEAAAgALAAAAAAjABQAQARZEMlJq724LrA7+IvHcUtmnmgWhshatu77mnFtx2muWyEp
gr4OC9VbjWIk304ynMyYz6V0Sj09o7DlbVvLbZLCorFpFQJdnzTHK/6Bw1gedOWku6r4vH6PiAAA
Ow==

--Apple-Mail-5–263420395
Content-Type: text/plain;
              charset=ISO-8859-1;
              format=flowed
Content-Transfer-Encoding: quoted-printable

F=F6re denna text =E4r det en gif-bild.=

--Apple-Mail-5–263420395--
```



Från: Ola Flygt
Ämne: **Ett meddelande med text och bild**
Datum: må 14 sep 2009 11.48.06 GMT+02:00
Till: Ola Flygt

Hejsan, detta är en text.

Före denna text är det en gif-bild.

Pretty Good Privacy

- Philip R. Zimmerman is the creator of PGP
- PGP provides a confidentiality and authentication service that can be used for electronic mail and file storage applications

Why Is PGP Popular?

- It is available free on a variety of platforms
- Based on well known algorithms
- Wide range of applicability
- Not developed or controlled by governmental or standards organizations
- Now however also standardized (RFC 3156)

Operational Description

- PGP consist of five services:
  - Authentication
  - Confidentiality
  - Compression
  - E-mail compatibility
  - Segmentation

Authentication

- Combination of SHA-1 and RSA provides an effective digital signature scheme
  - Because of the strength of RSA the recipient is assured that only the possessor of the matching private key can generate the signature
  - Because of the strength of SHA-1 the recipient is assured that no one else could generate a new message that matches the hash code
  - As an alternative, signatures can be generated using DSS/SHA-1
  - Detached signatures are supported
  - Each person's signature is independent and therefore applied only to the document

Confidentiality

- Provided by encrypting messages to be transmitted or to be stored locally as files
  - In both cases the symmetric encryption algorithm CAST-128 may be used
  - Alternatively IDEA or 3DES may be used
  - The 64-bit cipher feedback (CFB) mode is used

In PGP each symmetric key is used only once

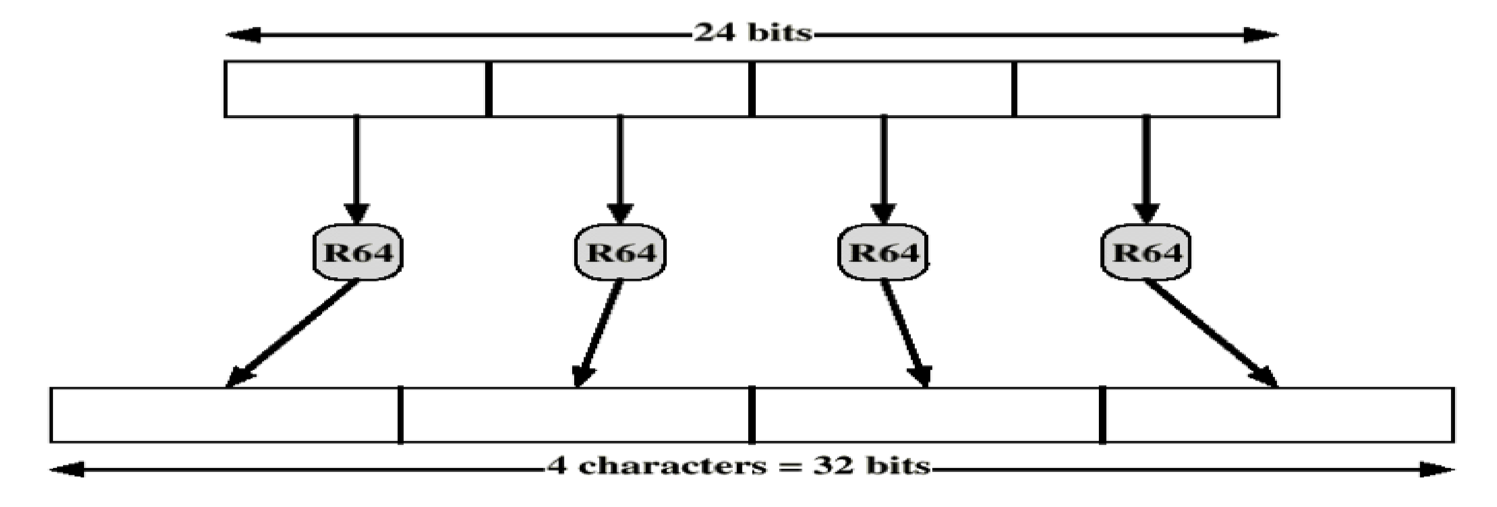
- As an alternative to the use of RSA for key encryption, PGP uses ElGamal, a variant of Diffie-Hellman that provides encryption/decryption

Compression

- PGP compresses the message after applying the signature but before encryption
- The placement of the compression algorithm is critical
- The compression algorithm used is ZIP (described in appendix G)

E-mail Compatibility

- The scheme used is radix-64 conversion (see appendix K)
- The use of radix-64 expands the message by 33%

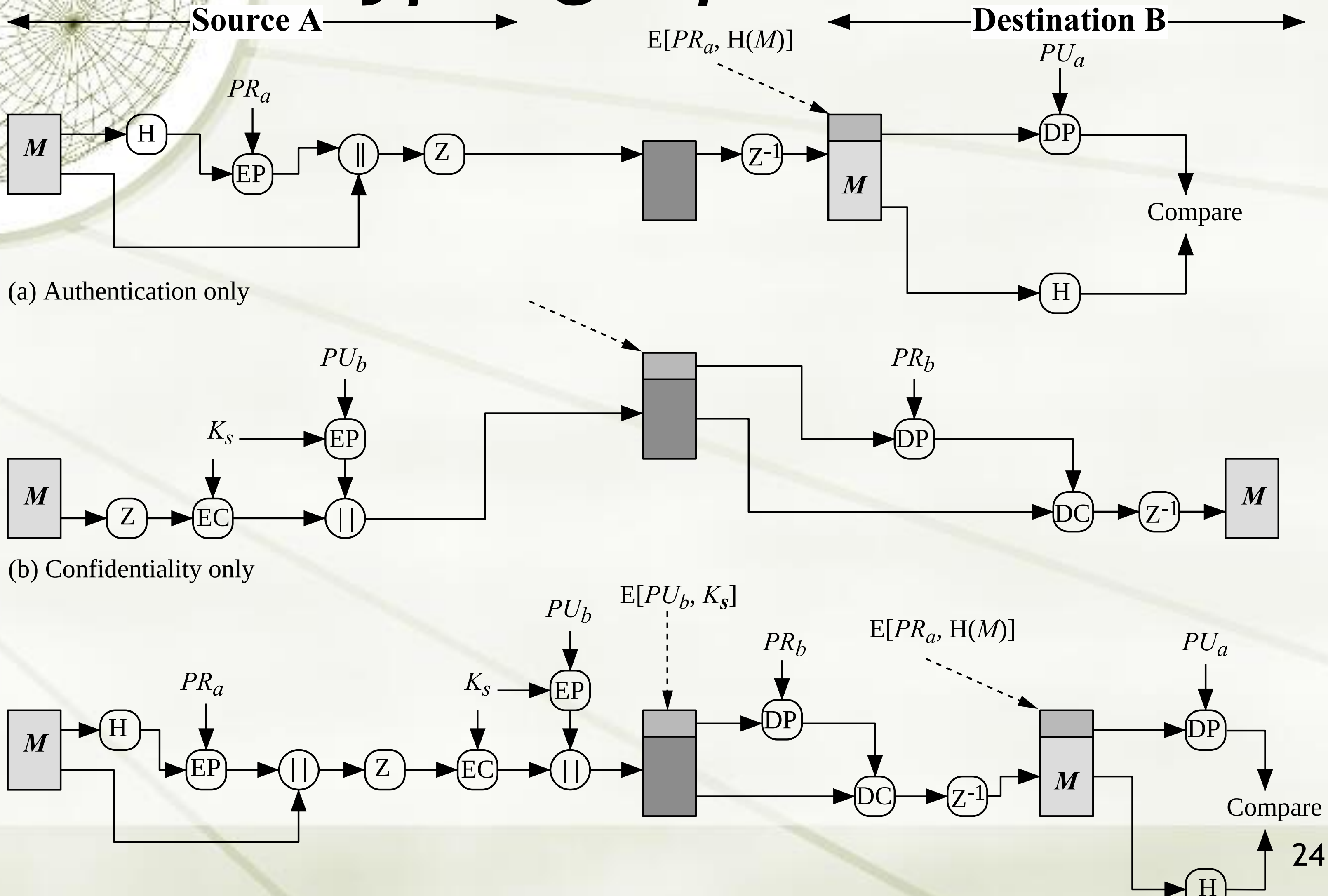


Check out https://www.cse.ust.hk/faculty/cding/CSIT571/SLIDES/Radix-64.pdf for details

Segmentation and Reassembly

- Internet Email delivery often restricted to a maximum message length of 50,000 octets
- Longer messages must be broken up into segments
- PGP automatically subdivides a message that is to large
- The receiver strip of all e-mail headers and reassemble the block

PGP Cryptographic Functions

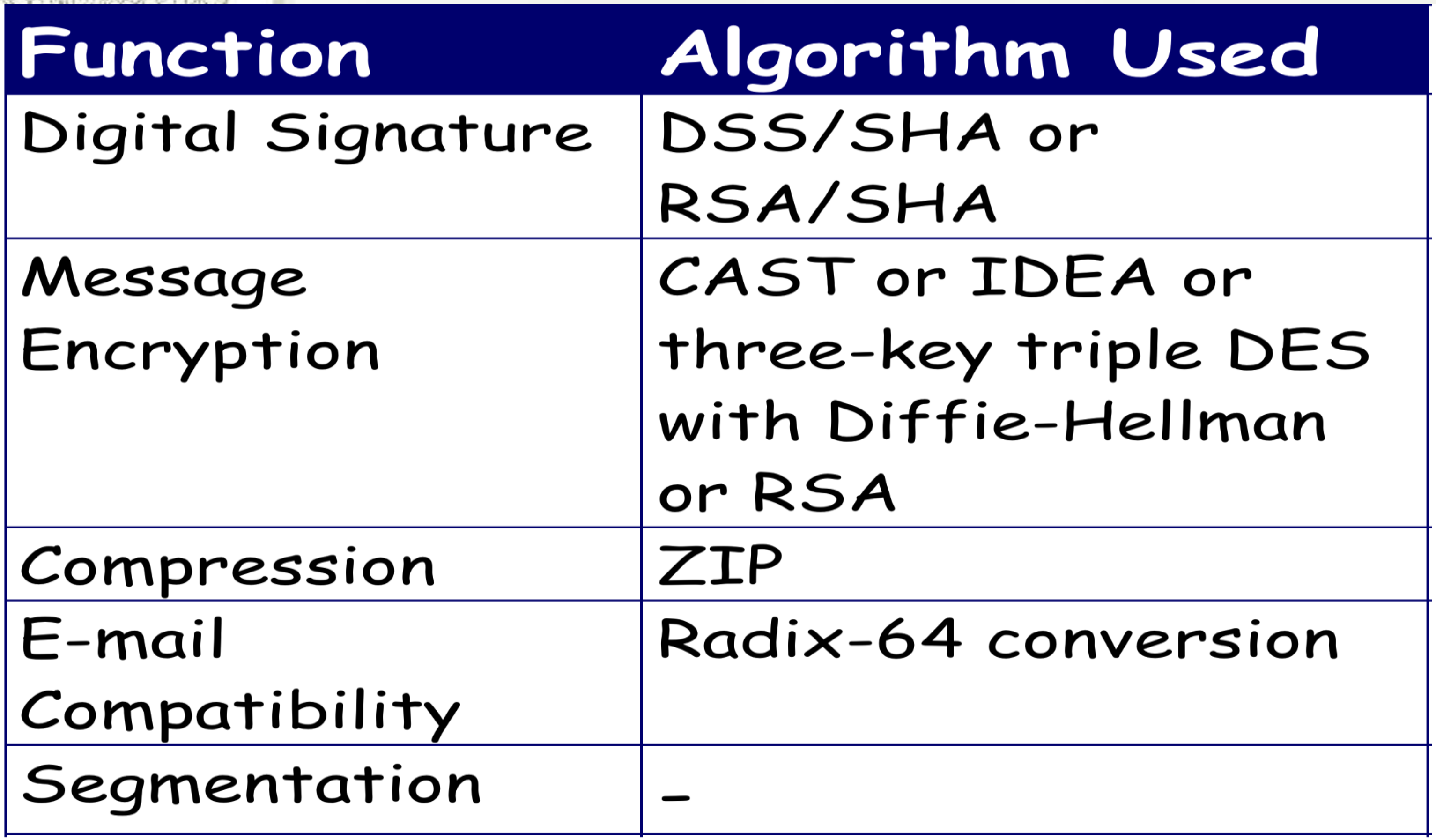


(a) Authentication only

(b) Confidentiality only

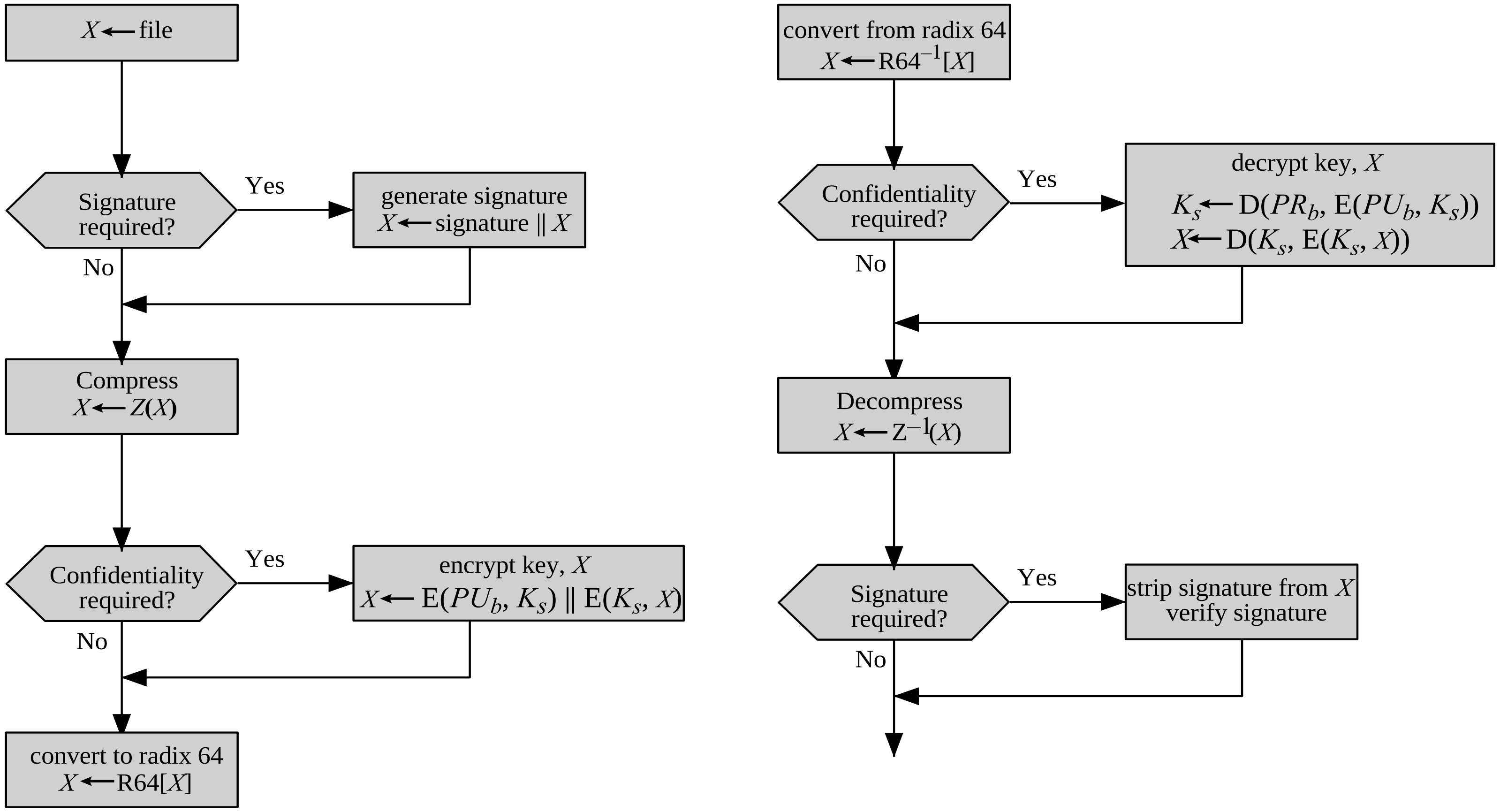
(c) Confidentiality and authentication

Summary of PGP Services



Function	Algorithm Used
Digital Signature	DSS/SHA or RSA/SHA
Message Encryption	CAST or IDEA or three-key triple DES with Diffie-Hellman or RSA
Compression	ZIP
E-mail Compatibility	Radix-64 conversion
Segmentation	-

PGP Operation – Summary



$X \leftarrow$ file

Signature required?

Yes

generate signature
$X \leftarrow$ signature || X

No

Compress
$X \leftarrow Z(X)$

Confidentiality required?

Yes

encrypt key, X
$X \leftarrow \mathrm{E}(PU_b, K_s) \parallel \mathrm{E}(K_s, X)$

No

convert to radix 64
$X \leftarrow \mathrm{R64}[X]$

(a) Generic Transmission Diagram (from A)

convert from radix 64
$X \leftarrow \mathrm{R64}^{-1}[X]$

Confidentiality required?

Yes

decrypt key, X
$K_s \leftarrow \mathrm{D}(PR_b, \mathrm{E}(PU_b, K_s))$
$X \leftarrow \mathrm{D}(K_s, \mathrm{E}(K_s, X))$

No

Decompress
$X \leftarrow \mathrm{Z}^{-1}(X)$

Signature required?

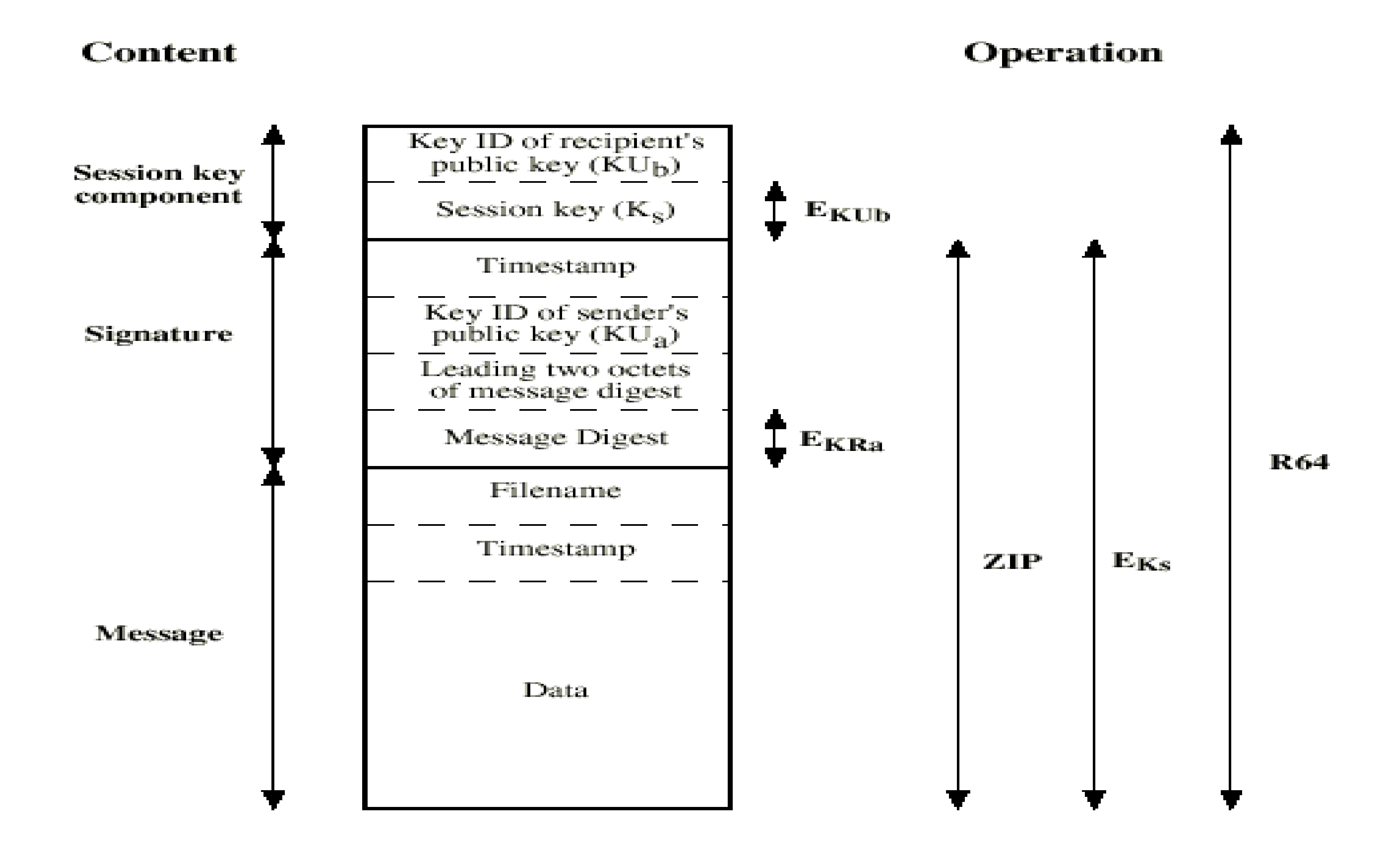
Yes

strip signature from X
verify signature

No

(b) Generic Reception Diagram (to B)

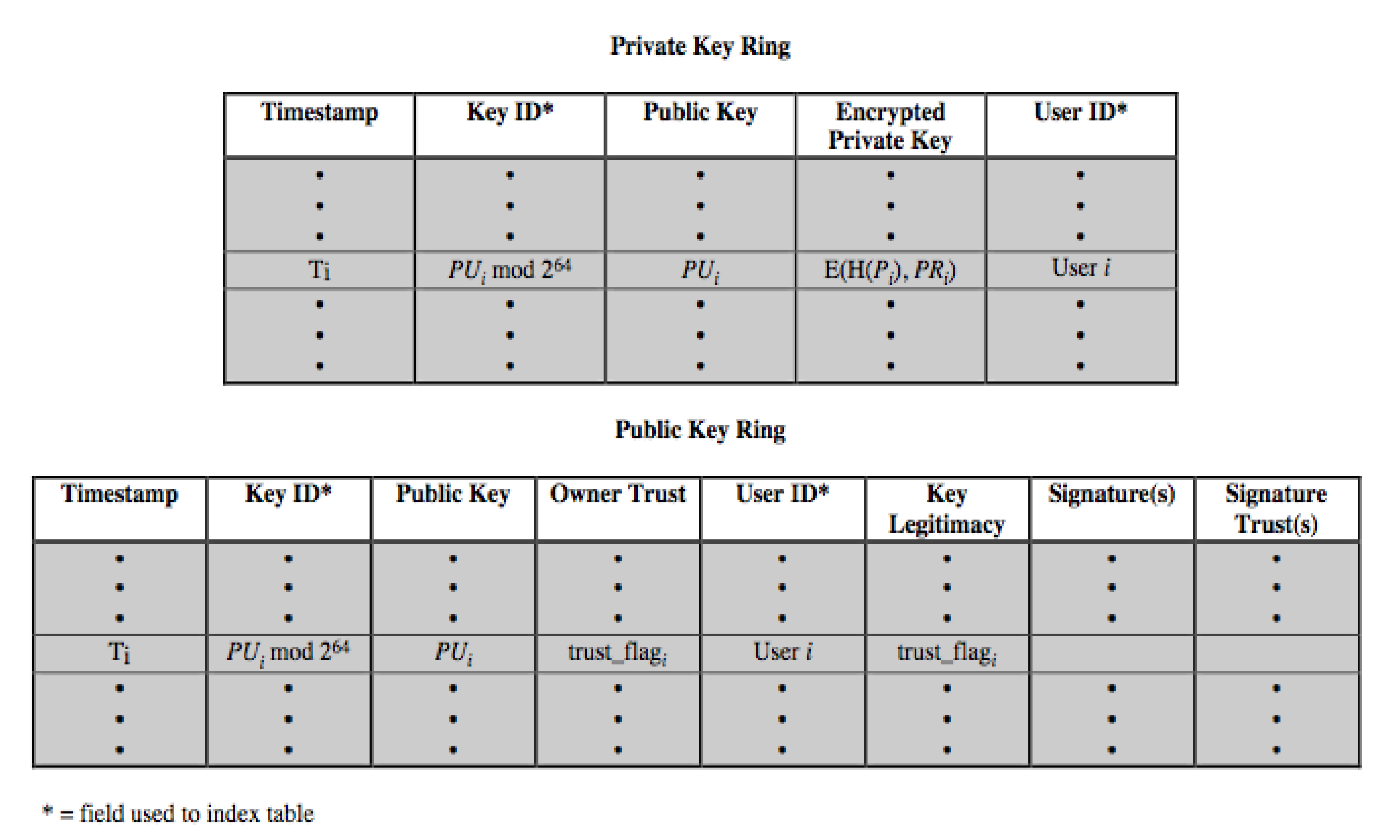
Format of PGP Message



PGP Key Rings

- each PGP user has a pair of keyrings:
  - public-key ring contains all the public-keys of other PGP users known to this user, indexed by key ID
  - private-key ring contains the public/private key pair(s) for this user, indexed by key ID & encrypted keyed from a hashed passphrase
- security of private keys thus depends on the pass-phrase security

PGP Key Rings



Private Key Ring

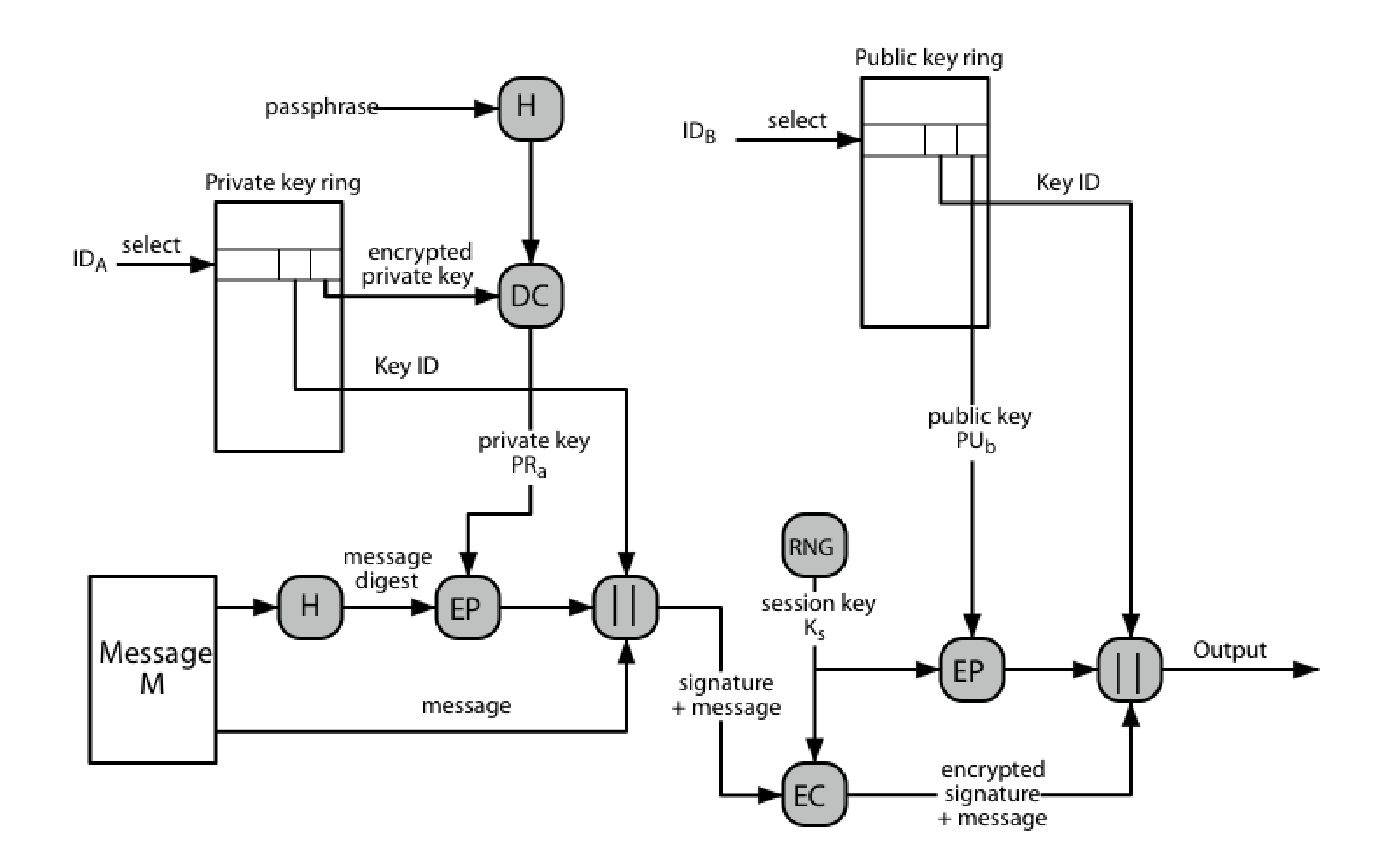
Timestamp	Key ID*	Public Key	Encrypted Private Key	User ID*
• • •	• • •	• • •	• • •	• • •
Ti	PU_i mod 2^{64}	PU_i	E(H(P_i), PR_i)	User *i*
• • •	• • •	• • •	• • •	• • •

Public Key Ring

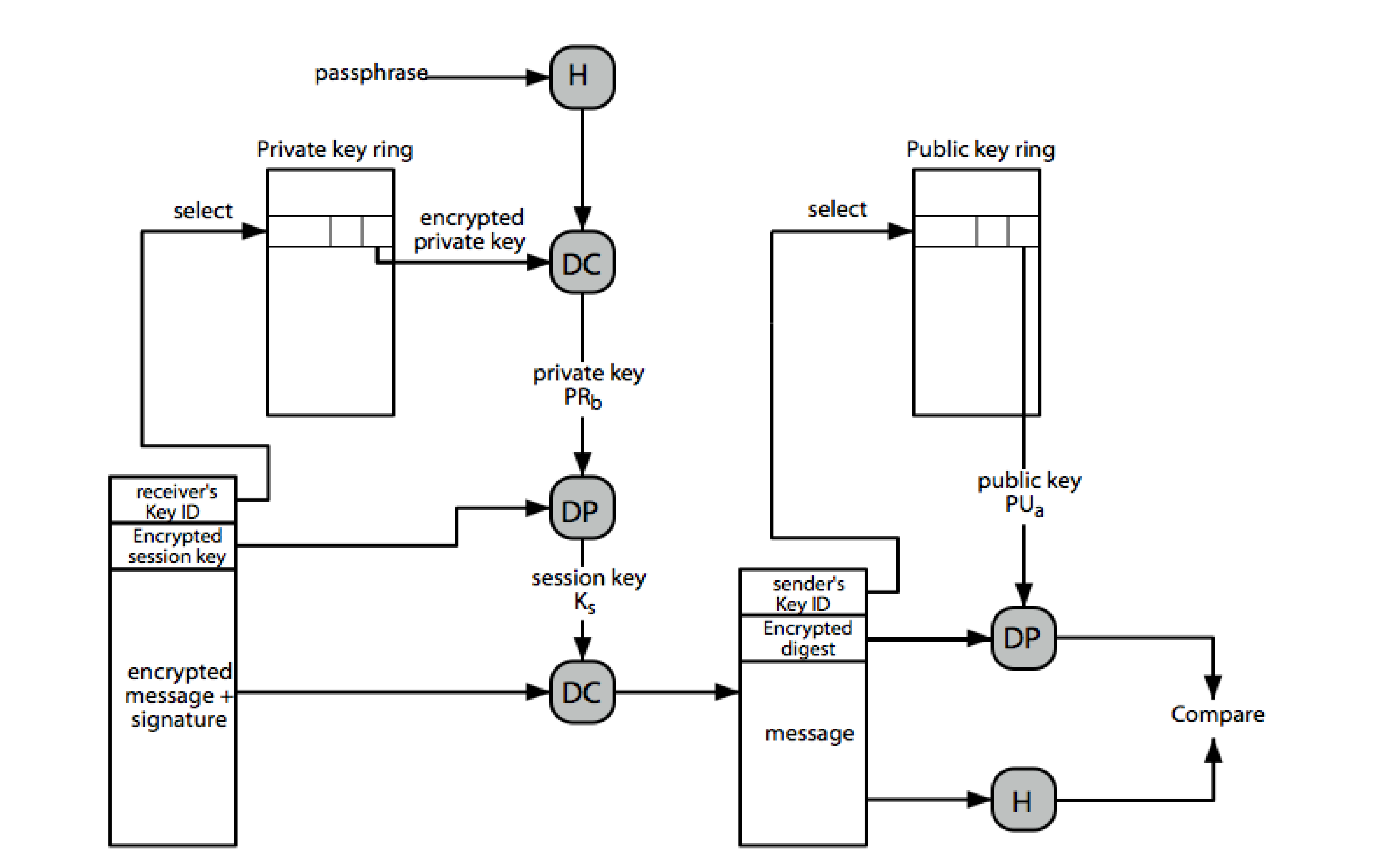
Timestamp	Key ID*	Public Key	Owner Trust	User ID*	Key Legitimacy	Signature(s)	Signature Trust(s)
• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •
Ti	PU_i mod 2^{64}	PU_i	trust_flag$_i$	User *i*	trust_flag$_i$		
• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •

* = field used to index table

PGP Message Generation



PGP Message Reception

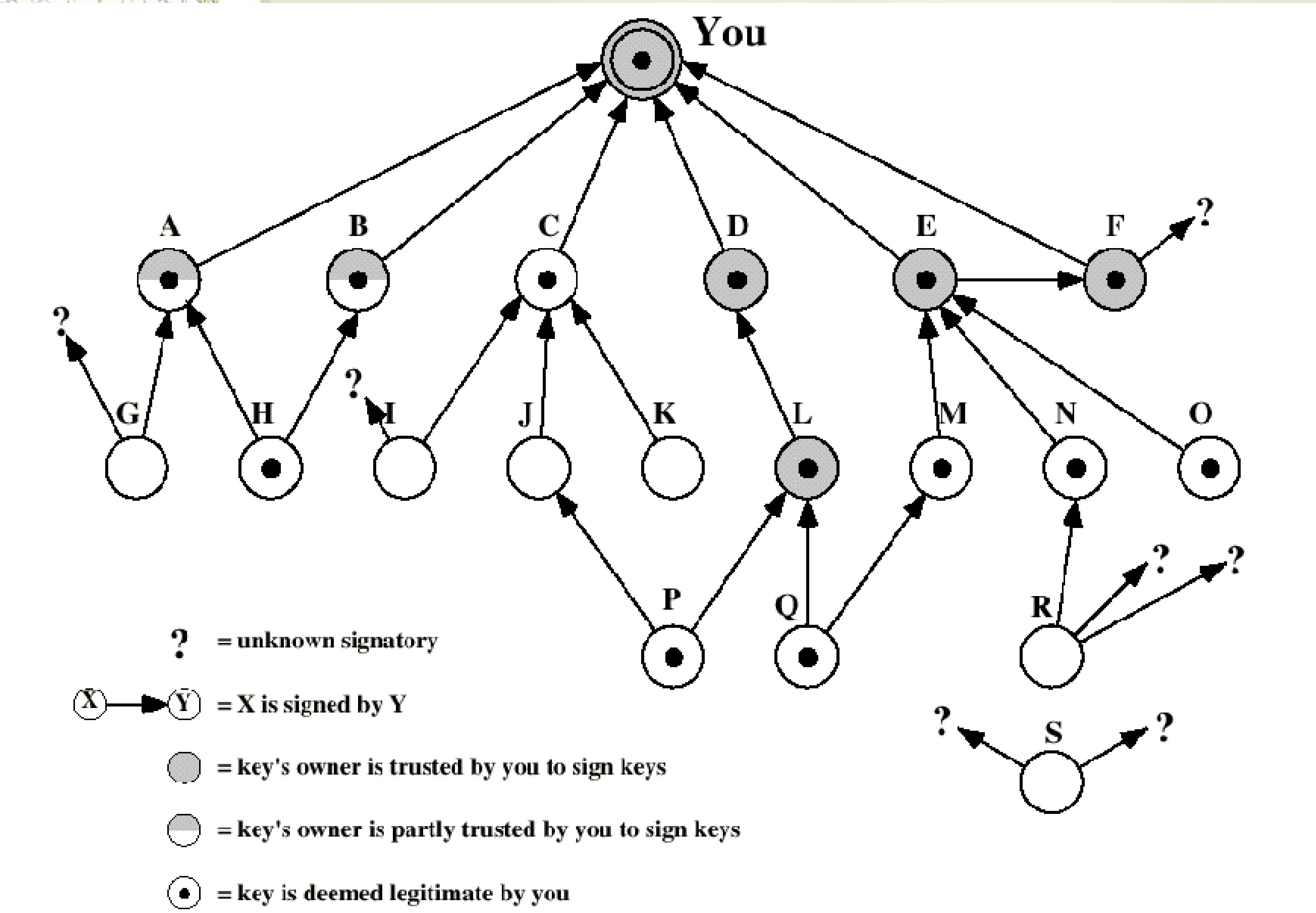


Obtaining trust

- How can we obtain trust in the the other parties public key?
  - Certificate
  - Getting the key personally
- PGP tries to give trust in a new way
  - Web of trust

The Use of Trust

- The user maintains a data structure with certified keys together with the following fields
  - Owner trust field, assigned by user
  - Signature trust field, cached copies of respective owner trust fields
  - Key legitimacy field, calculated by PGP
- Trust for each key is calculated as weighted sum of trust in signatures for this key



? = unknown signatory

X → Y = X is signed by Y

● = key's owner is trusted by you to sign keys

◒ = key's owner is partly trusted by you to sign keys

⊙ = key is deemed legitimate by you

Revoking Public Keys

- The owner issue a key revocation certificate.
  - Normal signature certificate with a revoke indicator.
  - Corresponding private key is used to sign the certificate

PGP Key servers

- Public key servers act as a phonebook for PGP keys, allowing a person to use an email address, name, or key fingerprint to search for a full key and download it
- There are many PGP public key servers, but they usually share their key collections with each other
- Key servers can't verify whether the keys they publish are genuine or forgeries. Anyone can upload a key to a public key server—in anyone's name
- In order to check the authenticity of a key, you need to check its signatures, or confirm its fingerprint with the original user in a trustworthy way
- PGP allows you to sign other people's keys, which is a way of using your own key to assert that a certain key is the right one to use to contact another person. This creates the web of trust

Key Servers poisoning

- In June 2019 an attack was made on the Key Server infrastructure
- Several keys were spammed by being signed thousands of times
- The information added to a Key Server can not be removed (for security reasons) but in this case it means that people syncing with these poisoned servers can crash due to the size of the signed keys
- There is still no remedy for this problem

Source: https://gist.github.com/rjhansen/67ab921ffb4084c865b3618d6955275f

S/MIME

- Secure/Multipurpose Internet Mail Extension (currently version 4.0, 2019)
- Defined in RFC 8551
- S/MIME was intended to become the industry standard for companies and other large organisations
- PGP for personal e-mail security

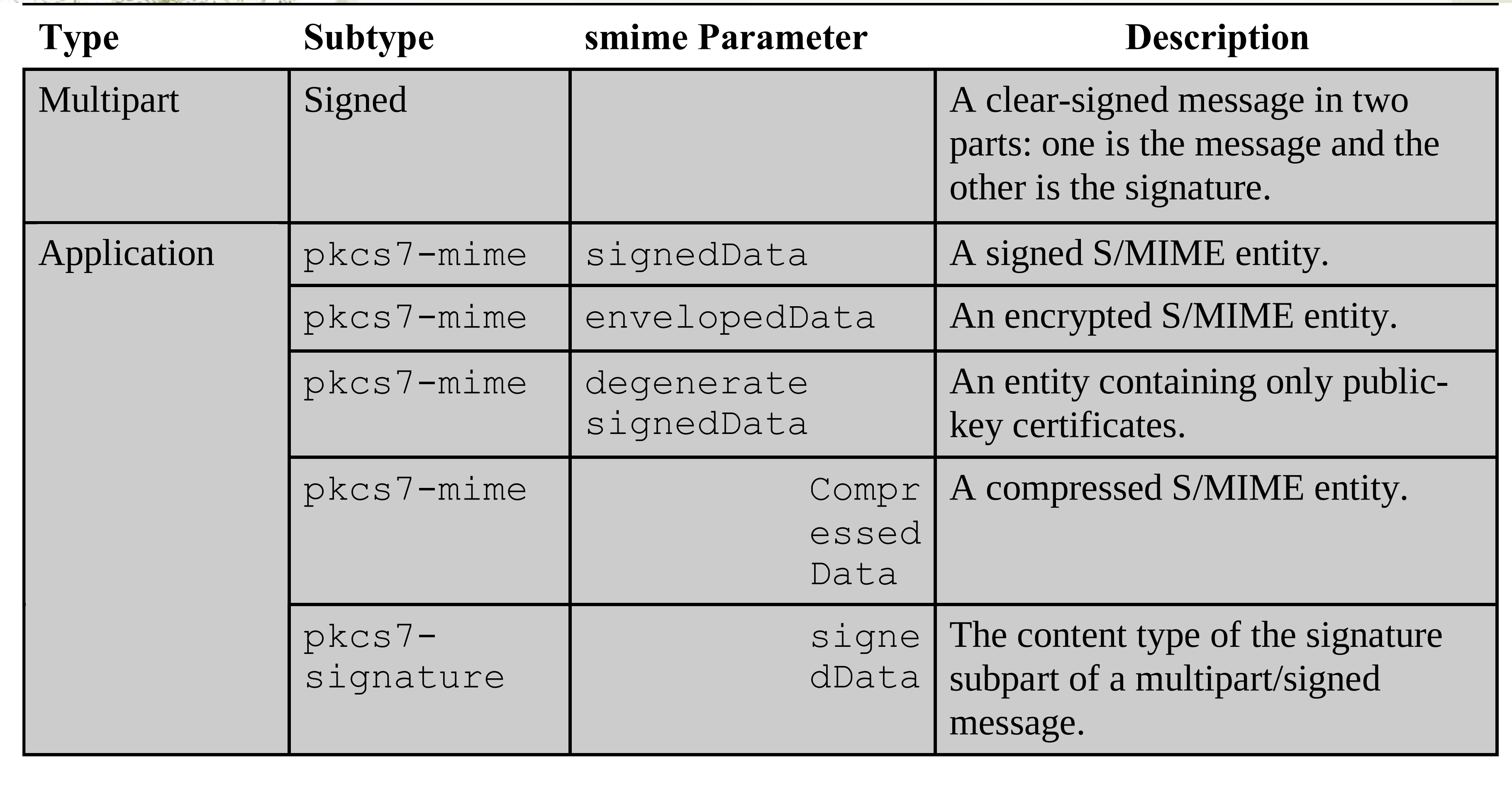
S/MIME Functions

- **Enveloped Data:** Encrypted content and encrypted session keys for recipients.
- **Signed Data:** Message Digest encrypted with private key of the signer and then content + signature is encoded using base64 encoding.
- **Clear-Signed Data:** Signed but not encoded except the signature.
- **Signed and Enveloped Data:** Various orderings for encrypting and signing.

Algorithms Used

- **Message Digesting:** SHA-1 and SHA-256
- **Digital Signatures:** RSA and DSA (DSS)
- **Secret-Key Encryption:** AES, Triple-DES, RC2/40
- **Public-Private Key Encryption:** RSA with key sizes of 512 and 1024 bits, and Diffie-Hellman (for session keys).

S/MIME Content 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.

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

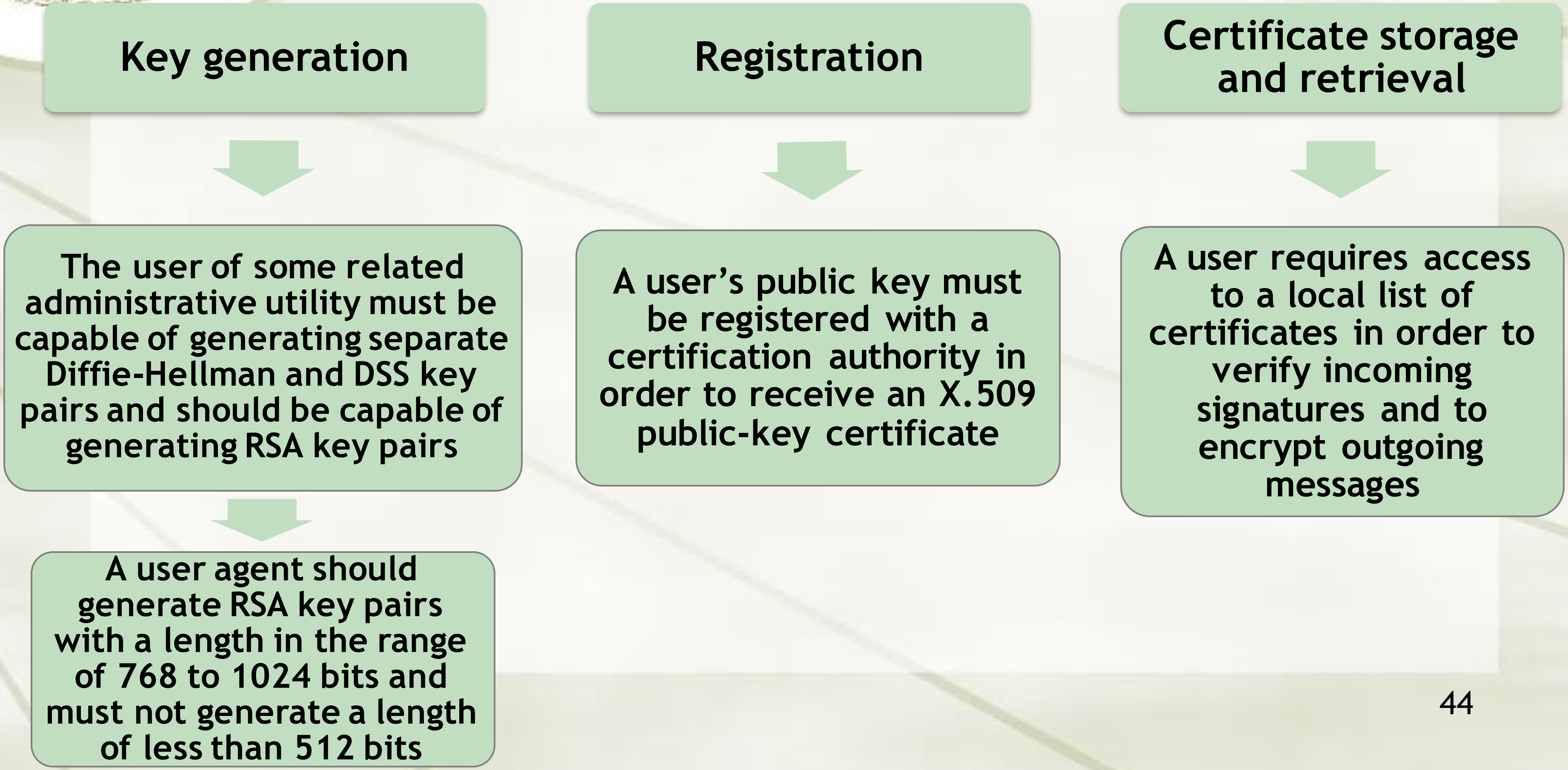
PKCS stands for "Public Key Cryptography Standards”

S/MIME Certificate Processing

- S/MIME uses public-key certificates that conform to version 3 of X.509
- The key-management scheme used by S/MIME is in some ways a hybrid between a strict X.509 certification hierarchy and PGP’s web of trust
- 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

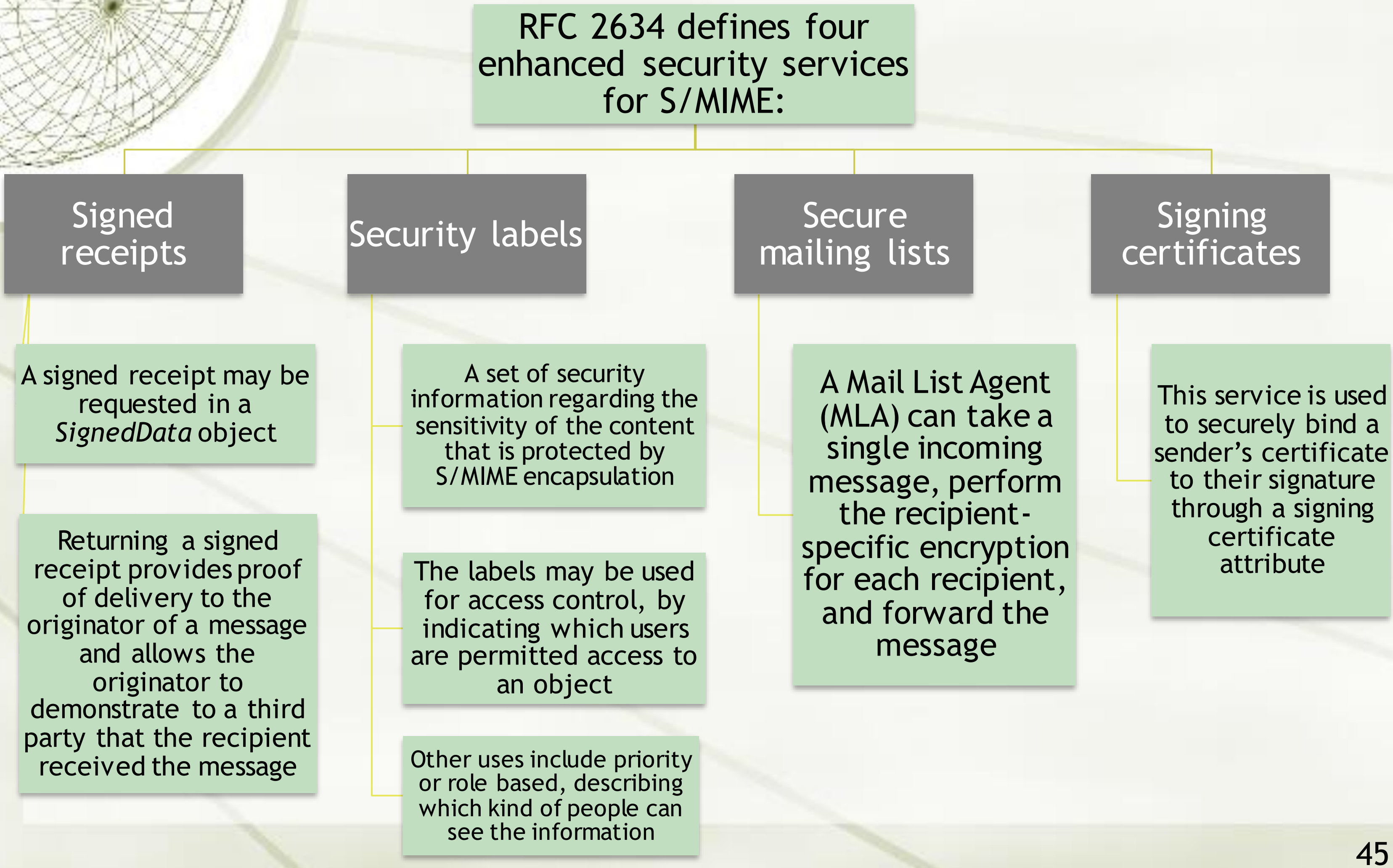
Registration

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 for S/MIME



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
- Security labels
  - 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 or role based, describing which kind of people can see the information
- Secure mailing lists
  - A 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

Efail – an example of an email vulnerability

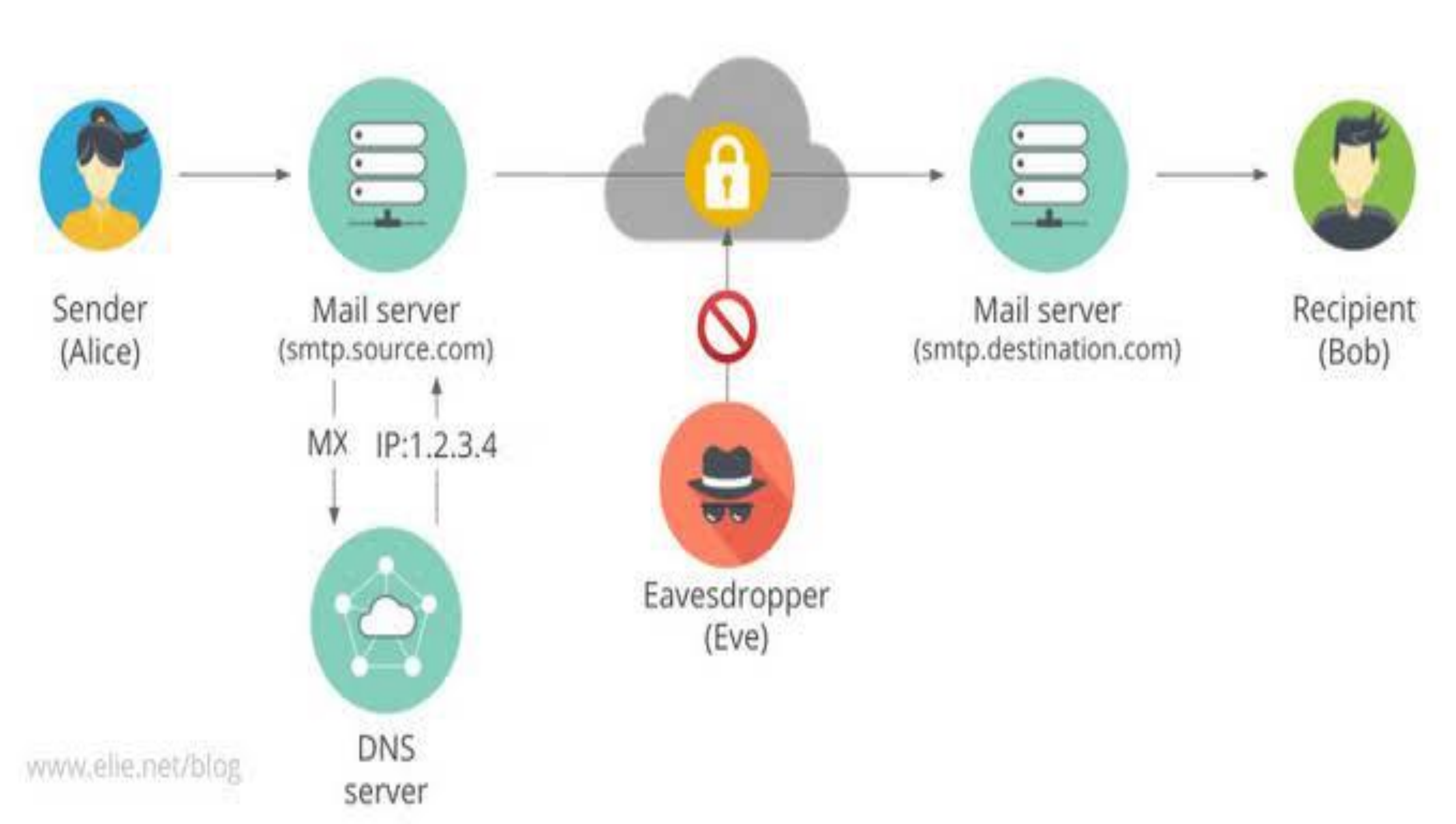
- Efail, is a security hole in email systems with which content can be transmitted in encrypted form was revealed in 2018. This gap allows attackers to access the decrypted content of an email if it contains active content like HTML or JavaScript, or if loading of external content has been enabled in the client. Affected email clients include Gmail, Apple Mail, and Microsoft Outlook and affect both PGP and S/MIME.

Source: https://en.wikipedia.org/wiki/EFAIL

Other problems with email

- Both PGP and S/MIME handles confidentiality, authentication of sender and integrity for e-mails
- Other security problems exist, e.g.
  - Spam
  - Phishing
  - Spoofing

Email delivery on internet



SMTP Strict Transport Security (SMTP STS)

- Top email providers, namely Google, Microsoft, Yahoo!, Comcast, LinkedIn, and 1&1 Mail & Media Development, have joined forces to develop a new email standard that makes sure the emails you send are going through an encrypted channel and cannot be sniffed.
- The primary goal of SMTP STS is to prevent Man-in-the-Middle (MitM) attacks that have compromised past efforts like STARTTLS at making SMTP a more secure protocol.

Reference: http://thehackernews.com/2016/03/smtp-sts-email-security.html

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

SMTP STS vs. DANE

- SMTP STS does not require DNSSEC (but it is recommended)
- SMTP STS defines a policy cache in the mail server
- SMTP STS requires x509 certificates that validate against a root-CA-certificate (no "self-signed" certs)
- SMTP STS requires a HTTPS server to serve a policy JSON document
- SMTP STS requires validation of the HTTPS connection to fetch the policy document

SMTP STS vs. DANE

- DANE does require DNSSEC
- DANE has no policy cache (but the TTL on TLSA records can work as such)
- DANE allows "self-signed" certificates
- DANE policy can be changed by switching the TLSA record in DNS
- DANE TLS-cert rollover need to be in sync with TLSA record(s)
- DANE relies on the trust on the DNSSEC chain

DomainKeys Identified Mail

- DomainKeys Identified Mail (DKIM) lets an organization take responsibility for a message while it is in transit.
- The organization is a handler of the message, either as its originator or as an intermediary. Their reputation is the basis for evaluating whether to trust the message for delivery.
- Technically DKIM provides a method for validating a domain name identity that is associated with a message through cryptographic authentication.

DomainKeys Identified Mail

- Protocol for signing and verifying the originating domain for a message in transit
- Prevents domain-level forgery
- Helps deal with spam and phishing
  - Increase effectiveness of blacklists
  - Ensure the identity of an sender domain
- Backwards compatible
  - Works with all existing MTAs and MUAs

DKIM Goals

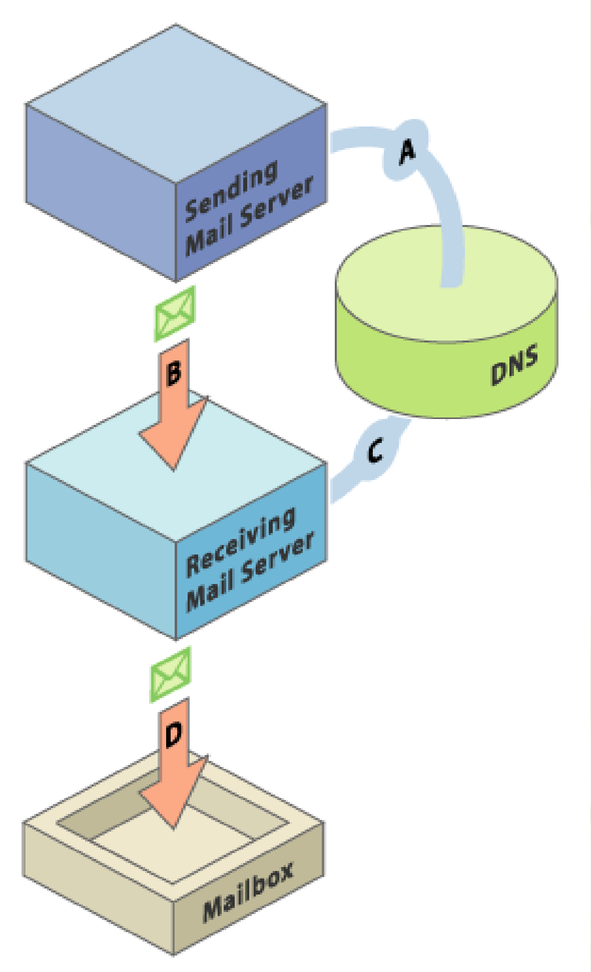
- Based on message content, itself
  - Not related to path
- Transparent to end users
  - No client User Agent upgrades *required*
  - But extensible to per-user signing
- Allow signature delegation
  - Outsourcing
- Low development, deployment, use costs
  - Avoid large PKI, new Internet services
  - No trusted third parties (except DNS)

Technical High-points

- Signs body and selected parts of header
- Signature transmitted in DKIM-Signature: header
- Public key stored in DNS
  - In _domainkey subdomain
  - Uses TXT RR (see http://en.wikipedia.org/wiki/List_of_DNS_record_types)
- Namespace divided using selectors
  - Allows multiple keys for aging, delegation, etc.

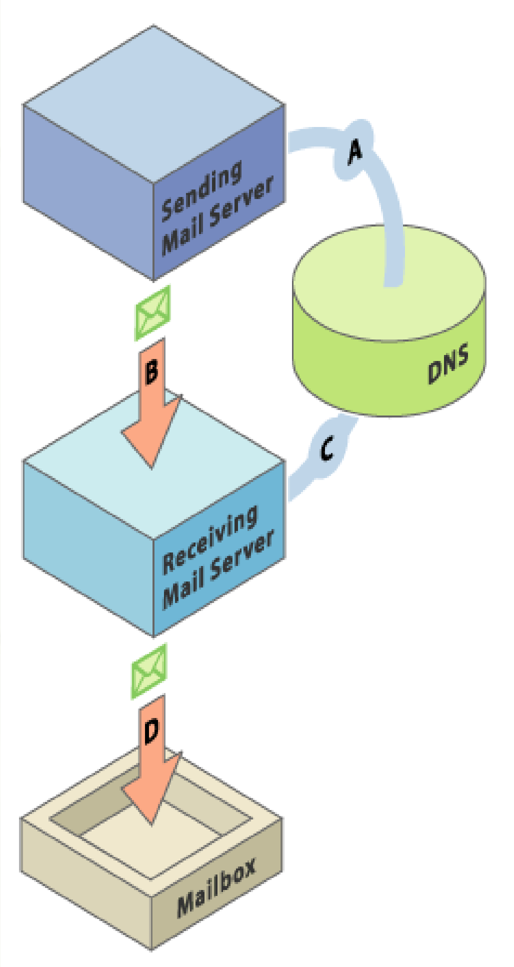
How it works – Sending Servers

- Set up: The domain owner (typically the team running the email systems within a company or service provider) generates a public/private key pair to use for signing all outgoing messages (multiple key pairs are allowed). The public key is published in DNS, and the private key is made available to their DomainKey-enabled outbound email servers. This is step "A" in the diagram to the right.
- Signing: When each email is sent by an authorized end-user within the domain, the DomainKey-enabled email system automatically uses the stored private key to generate a digital signature of the message. This signature is then pre-pended as a header to the email, and the email is sent on to the target recipient's mail server. This is step "B" in the diagram to the right.



How it works – Receiving Servers

- Preparing: The DomainKeys-enabled receiving email system extracts the signature and claimed From: domain from the email headers and fetches the public key from DNS for the claimed From: domain. This is step "C" in the diagram to the right.
- Verifying: The public key from DNS is then used by the receiving mail system to verify that the signature was generated by the matching private key. This proves that the email was truly sent by, and with the permission of, the claimed sending From: domain and that its headers and content weren't altered during transfer.
- Delivering: The receiving email system applies local policies based on the results of the signature test. If the domain is verified and other anti-spam tests don't catch it, the email can be delivered to the user's inbox. If the signature fails to verify, or there isn't one, the email can be dropped, flagged, or quarantined. This is step "D" in the diagram on the right.



DomainKeys Identified Mail

- Example Signature Header

```
DKIM-Signature: v=1; a=rsa-sha256; d=example.net; s=brisbane;
c=relaxed/simple; q=dns/txt; t=1117574938; x=1118006938;
h=from:to:subject:date:keywords:keywords;
bh=MTIzNDU2Nzg5MDEyMzQ1Njc4OTAxMjM0NTY3ODkwMTI=;
b=dzdVyOfAKCdLXdJOc9G2q8LoXSlEniSbav+yuU4zGeeruD00lszZVoG4ZHRNiYzR
```

- Where the tags used are:

v, version
a, signing algorithm
d, domain
s, selector
c, canonicalization algorithm(s) for header and body
q, default query method

t, signature timestamp
x, expire time
h, header fields - list of those that have been signed
bh, body hash
b, signature of headers and body

- Note that the DKIM-Signature header field itself is always implicitly included in **h**.

For more information: https://en.wikipedia.org/wiki/DomainKeys_Identified_Mail

DomainKeys Identified Mail

- Already in the RFC they mentioned possible attacks on DKIM like Misuse of body length limits and Misappropriated private keys. No major problems have been found however.
- In 2017, another working group was launched, DKIM Crypto Update (dcrup), with the specific restriction to review signing techniques.
  - RFC 8301 was issued in January 2018. It bans SHA-1 and updates key sizes (from 512-2048 to 1024-4096).
  - RFC 8463 was issued in September 2018. It adds an elliptic curve algorithm to the existing RSA.

Email Deniability

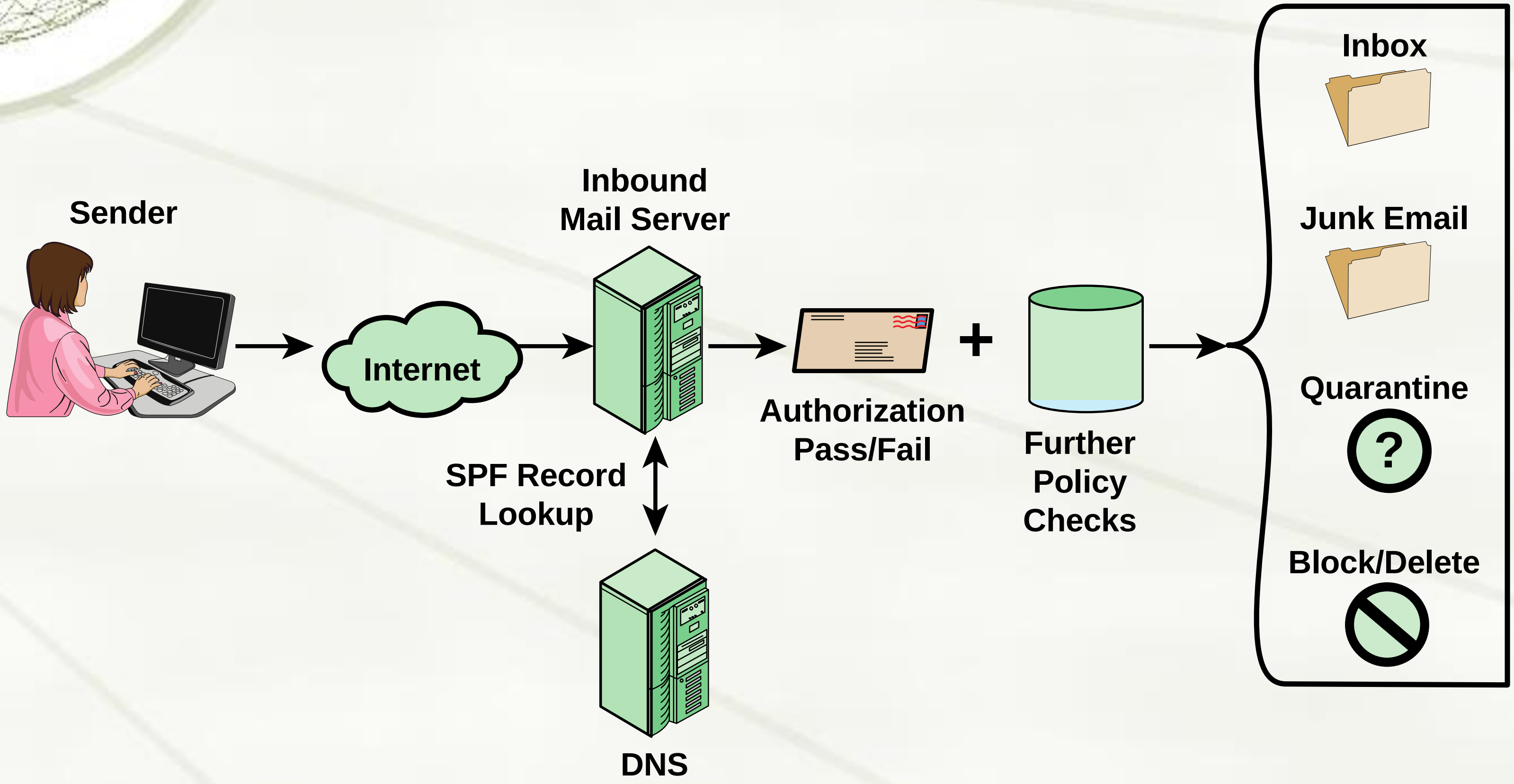
- PGP and S/MIME both add the option to sign messages, linking the message to you
- DKIM sign all outgoing emails which is very helpful to e.g. filter out spam and spoofing emails
- What if you want to be able to deny an email has been sent?
- The problem is that signing keys bind you to the email. What if the key easily could be forged, say after 15 min?
- KeyForge & TimeForge are solutions for this

Source: http://www.mit.edu/~specter/blog/2020/dkim/

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

Sender policy framework Operation



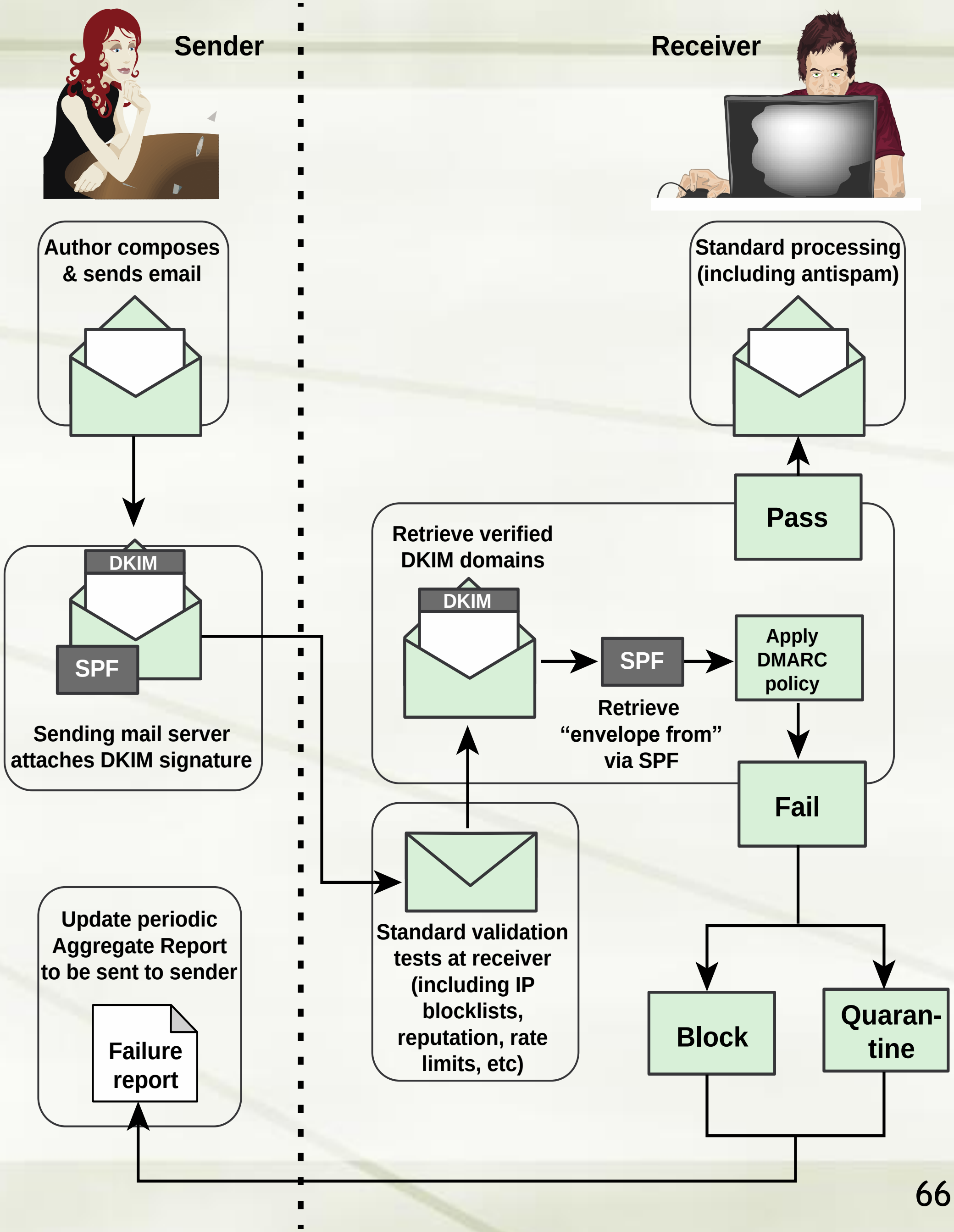
DMARC

- We looked at two mechanisms used to accomplish email authentication - DomainKeys identified mail (DKIM) and sender policy framework (SPF). These standards have been integrated into DMARC (Domain-based Messaging Authentication, Reporting and Conformance)
- Google, Yahoo, Microsoft and many others use it to enforce email security by aligning sender and recipient information
- Is defined in RFC 7489, March 2015

DMARK overview

- A DMARC policy allows a sender to indicate that their emails are protected by SPF and/or DKIM, and tells a receiver what to do if neither of those authentication methods passes - such as junk or reject the message
- It removes the guesswork from the receiver's handling of these failed messages, limiting or eliminating the user's exposure to potentially fraudulent & harmful messages
- It also provides a way for the email receiver to report back to the sender about messages that pass and/or fail DMARC evaluation

DMARK Functional Flow


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 processing (including antispam)
Pass
Retrieve verified DKIM domains
DKIM
SPF
Apply DMARC policy
Retrieve "envelope from" via SPF
Fail
Standard validation tests at receiver (including IP blocklists, reputation, rate limits, etc)
Block
Quaran-tine

DMARC debate

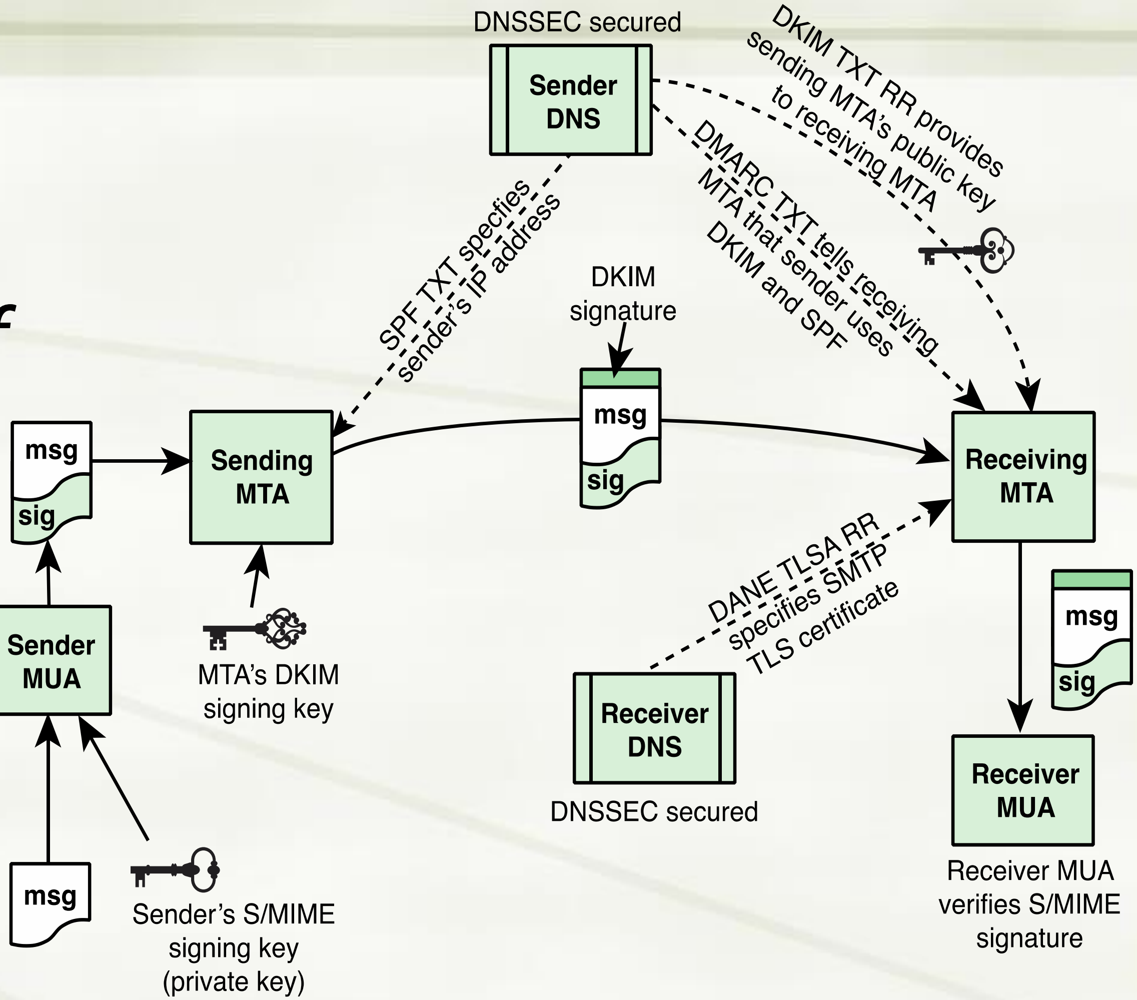
- In April 2014, Yahoo changed its DMARC policy to p=reject, thereby causing misbehaviour in several mailing lists. It was not yet a standard protocol and missed a provision for such sudden changes. This change caused significant debate in the IETF mailing list about whether DMARC was ready for deployment, about whether it is good for the Internet at all, and about whether its making process was correct.
- No similar problems have been seen after that and today it is regarded as a mature standard.

Use of SPF and DMARC

- An OTech study (March 2017) found that 86 percent of major online businesses are using Sender Policy Framework (SPF) to determine whether messages that claim to be from the businesses' email addresses actually come from the businesses.
- Fewer than 10 percent of the businesses, however, have implemented the supplemental technology Domain Message Authentication Reporting & Conformance (DMARC) in a manner which would allow the businesses to receive intelligence on potential spoofing attempts and to instruct ISPs to automatically reject any unauthenticated messages that claimed to be from the businesses' email addresses
- A study from 2022 show that 43.4% of domains in large businesses have a DMARC policy enforcement.

Sources:
https://www.ftc.gov/news-events/press-releases/2017/03/online-businesses-could-do-more-protect-their-reputations-prevent
https://godmarc.com/blogs/dmarc-and-email-statistics/

The Inter-relationship of DNSSEC, SPF, DKIM, DMARC, DANE and S/MIME for Assuring Message Authenticity and Integrity


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
Sender MUA
MTA's DKIM signing key
DANE TLSA RR specifies SMTP TLS certificate
Receiver DNS
DNSSEC secured
msg
sig
Receiver MUA
Receiver MUA verifies S/MIME signature
msg
Sender's S/MIME signing key (private key)

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