Ασφάλεια ηλεκτρονικού ταχυδρομείου

Νικόλαος Ε. Κολοκοτρώνης
Επίκουρος Καθηγητής

Τμήμα Πληροφορικής και Τηλεπικοινωνιών
Πανεπιστήμιο Πελοποννήσου
Email: nkolok@uop.gr
Web: http://www.uop.gr/~nkolok/

ΑΣΦΑΛΕΙΑ ΣΥΣΤΗΜΑΤΩΝ

Περιεχόμενα

I HAVE A NEW HOBBY. IT'S CALLED PHISHING.
I SEND FAKE BANKING E-MAILS TO GULLIBLE EXECUTIVES. THEN I FIND OUT THEIR FINANCIAL INFORMATION AND USE IT TO STEAL THE MONEY THEY DON'T DESERVE.
Dear Customer,
This is your bank. We forgot your social security number and password. Why don't you send them to us so we can protect your money.
Sincerely,
I. B. Banker
LOOKS LEGIT.
www.dilbert.com scottadams@aol.com
8-12-05 © 2005 Scott Adams, Inc./Dist. by UFS, Inc.

Περιεχόμενα

- Pretty good privacy
  - Notation
  - Operational description
- DomainKeys Identified Mail
  - Internet mail architecture
  - E-mail threats
  - DKIM strategy
  - DKIM functional flow
- S/MIME
  - RFC 5322
  - Multipurpose Internet mail extensions
  - S/MIME functionality
  - S/MIME messages
  - S/MIME certification processing
  - Enhanced security services

Email security

- email is one of the most widely used and regarded network services
- currently message contents are not secure
  - may be inspected either in transit
  - or by suitably privileged users on destination system

Email security enhancements

- Confidentiality
  - protection from disclosure
- Authentication
  - of sender of message
- Message integrity
  - protection from modification
- Non-repudiation of origin
  - protection from denial by sender

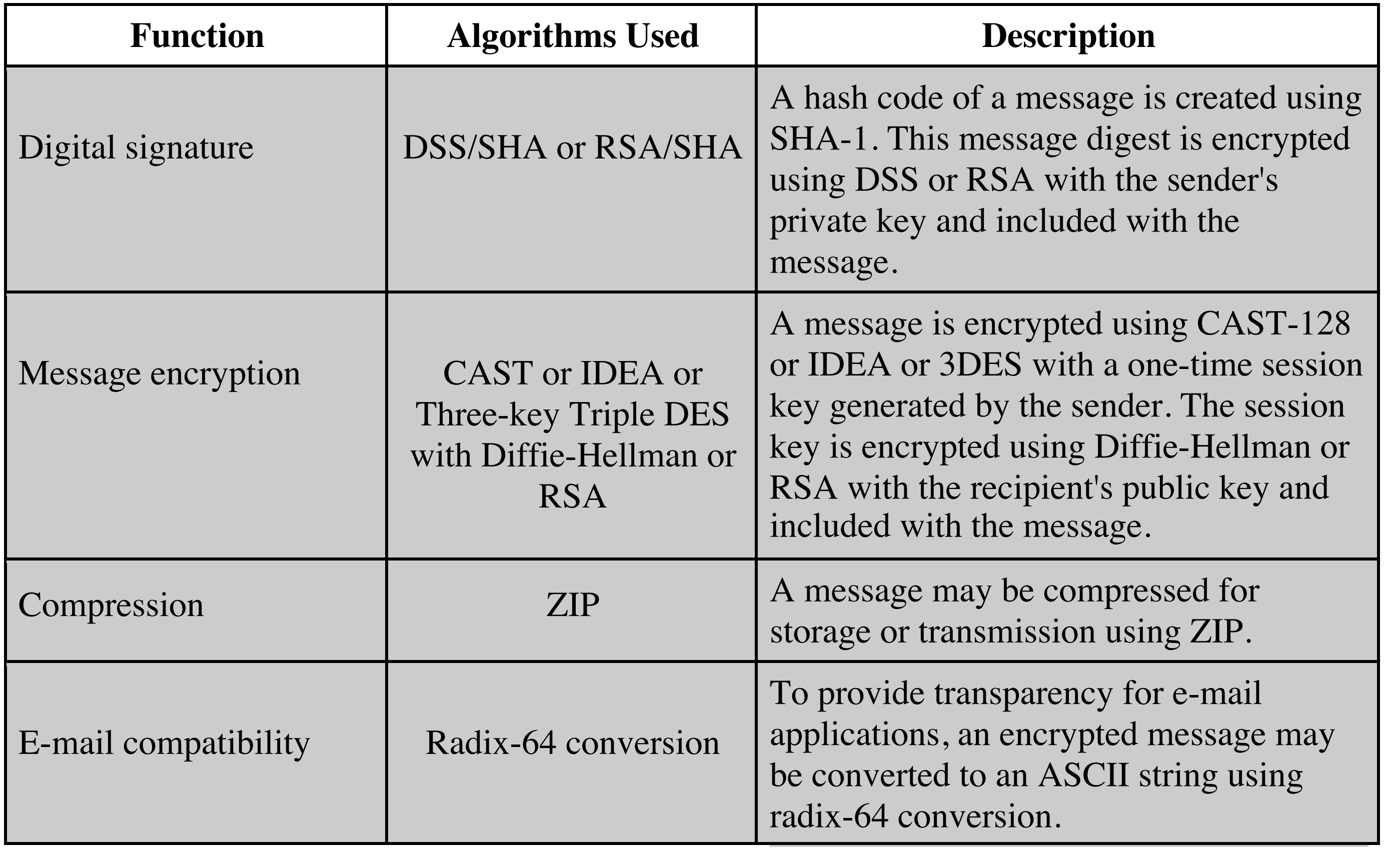
Pretty good privacy

- Developed by Phil Zimmermann
- Selected the best available cryptographic algorithms
  - Integrated into a general-purpose application that is based on a small set of easy-to-use commands
- Made the source code, package, and documentation, freely available on the Internet
- Entered into an agreement with a company to provide a fully compatible, low-cost commercial version of PGP

PGP growth

- It is available free worldwide in versions that run on a variety of platforms
- The commercial version satisfies users who want a product that comes with vendor support
- It is based on algorithms that have survived extensive public review and are considered extremely secure
- It has a wide range of applicability
- It was not developed by, nor is it controlled by, any governmental or standards organization
- Is now on an Internet standards track, however it still has an aura of an antiestablishment endeavor

Summary of PGP services

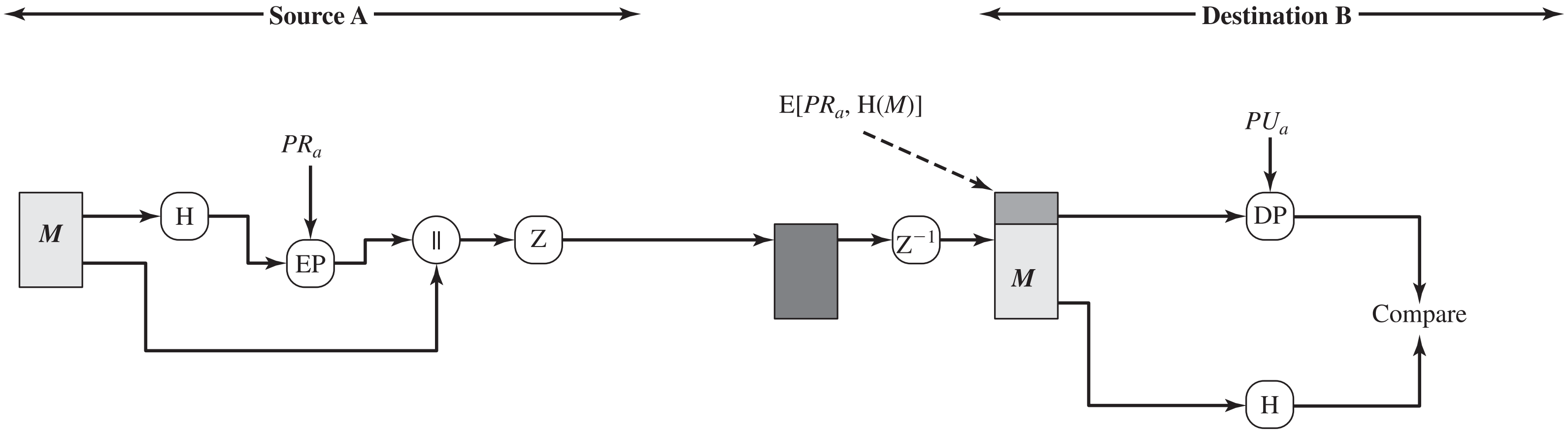


Function	Algorithms Used	Description
Digital signature	DSS/SHA or RSA/SHA	A hash code of a message is created using SHA-1. This message digest is encrypted using DSS or RSA with the sender's private key and included with the message.
Message encryption	CAST or IDEA or Three-key Triple DES with Diffie-Hellman or RSA	A message is encrypted using CAST-128 or IDEA or 3DES with a one-time session key generated by the sender. The session key is encrypted using Diffie-Hellman or RSA with the recipient's public key and included with the message.
Compression	ZIP	A message may be compressed for storage or transmission using ZIP.
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.

PGP authentication

- Combination of SHA-1 and RSA provides an effective digital signature scheme
  - The recipient is assured that only the possessor of the matching private key can generate the signature
  - 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
  - Each person’s signature is independent and therefore applied only to the document

PGP authentication



PGP confidentiality

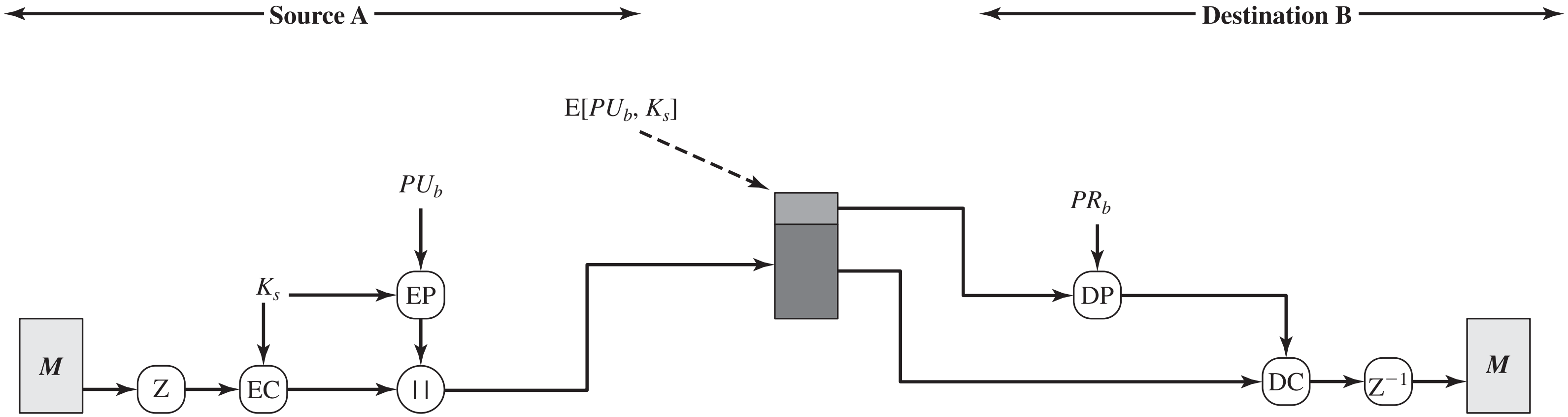
- Achieved by encrypting messages (CFB mode is used) to be transmitted or to be stored locally as files
  - The symmetric encryption algorithm CAST-128 may be used
  - Alternatively IDEA or 3DES may be used

In PGP each symmetric key is used only once

- Although referred to as a session key, it is in reality a one-time key
- Session key is bound to the message and transmitted with it
- To protect the key, it is encrypted with the receiver’s public key

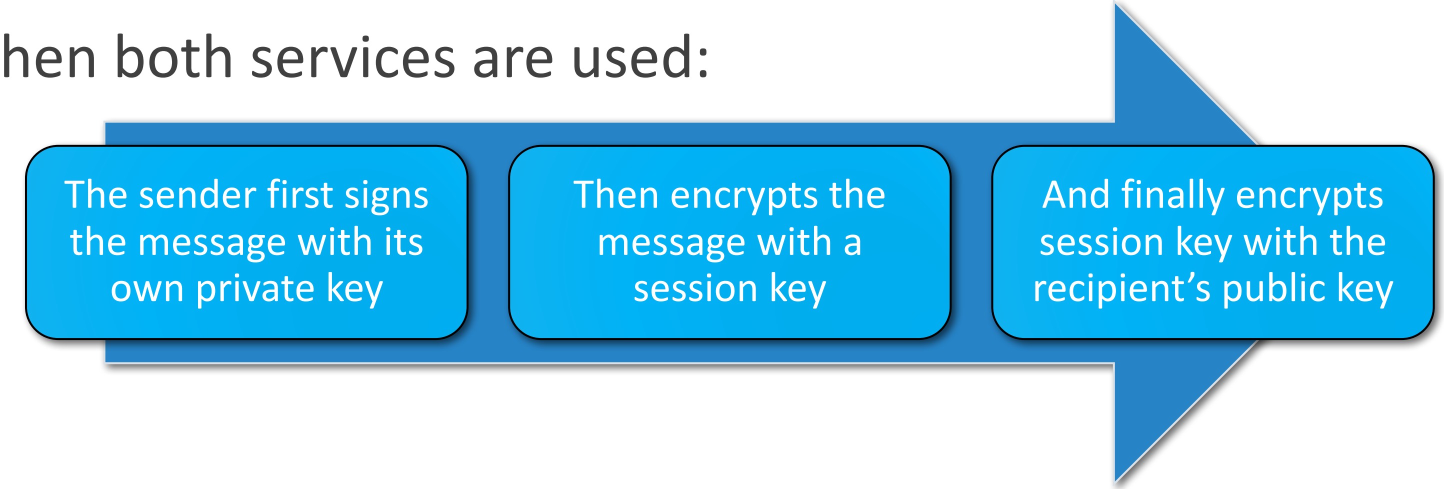
- Instead of RSA for key encryption, PGP can use ElGamal, a variant of DH that provides encryption/decryption

PGP confidentiality



PGP confidentiality and auth.

- Both services may be used for the same message
  - First a signature is generated for the plaintext message and prepended to the message
  - Then the plaintext + signature is encrypted using CAST-128 (or IDEA or 3DES) and the session key is encrypted using RSA (or ElGamal)
- When both services are used:

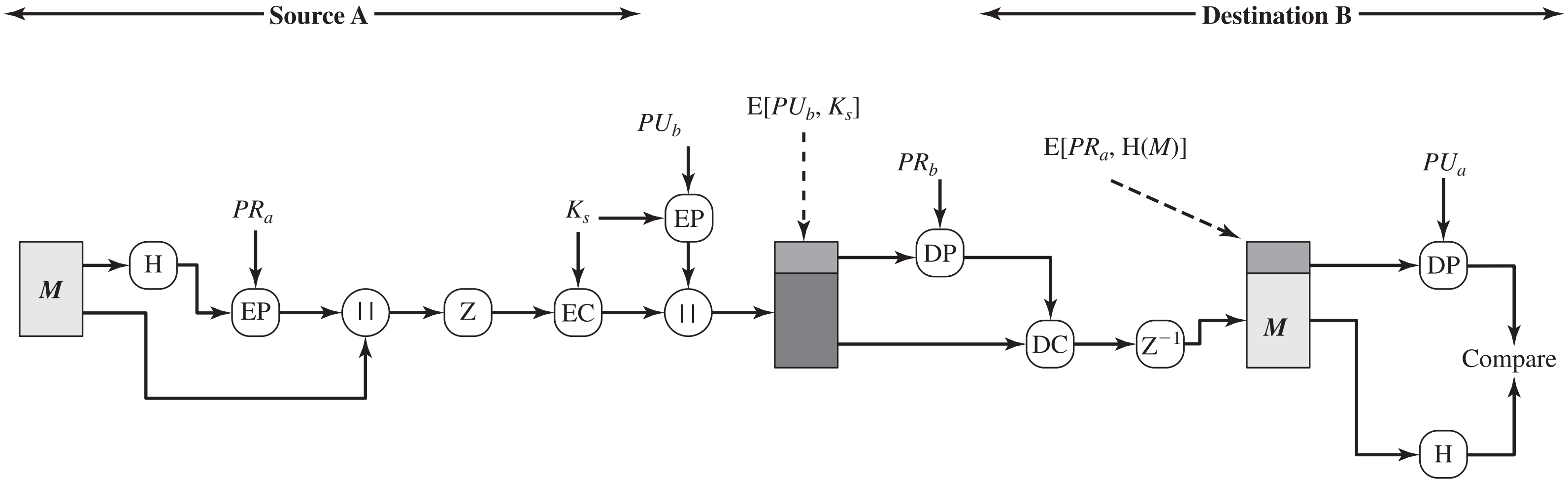


The sender first signs the message with its own private key

Then encrypts the message with a session key

And finally encrypts session key with the recipient’s public key

PGP confidentiality and auth.



PGP compression

- As a default, PGP compresses the message after applying the signature but before encryption
  - This has the benefit of saving space both for e-mail transmission and for file storage
  - The placement of the compression algorithm is critical
    - Applying the hash function and signature after compression would constrain all PGP implementations to the same version of the compression algorithm
    - Message encryption is applied after compression to strengthen cryptographic security
  - The compression algorithm used is ZIP

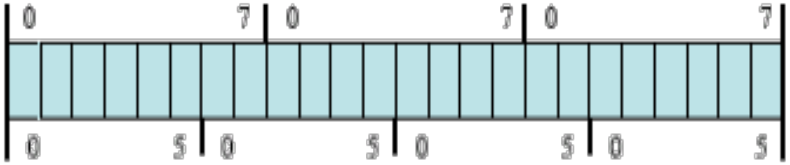
PGP e-mail compatibility

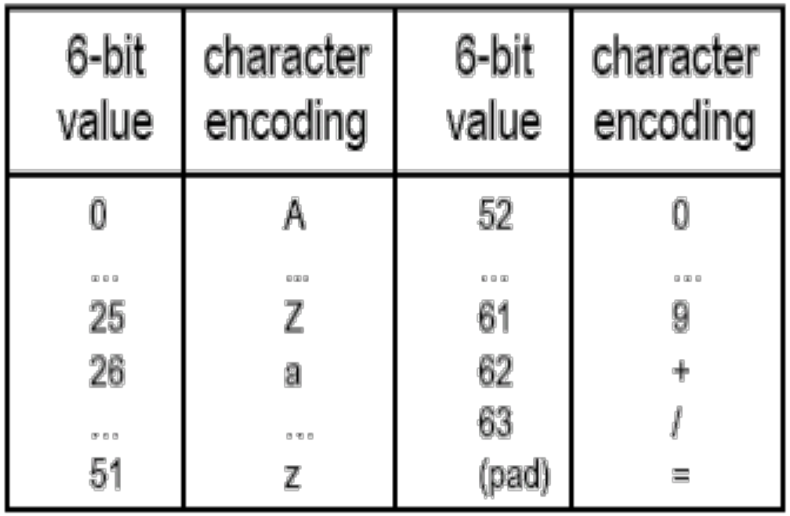
- Many electronic mail systems only permit the use of blocks consisting of ASCII text
    - To accommodate this restriction, PGP provides the service of converting the raw 8-bit binary stream to a stream of printable ASCII characters
    - The scheme used for this purpose is radix-64 conversion

PGP e-mail compatibility

RADIX-64 CONVERSION

- 3 x 8-bit blocks => 4 x 6-bit blocks
- Increases the file size 33.3%
- also appends a CRC to detect transmission errors

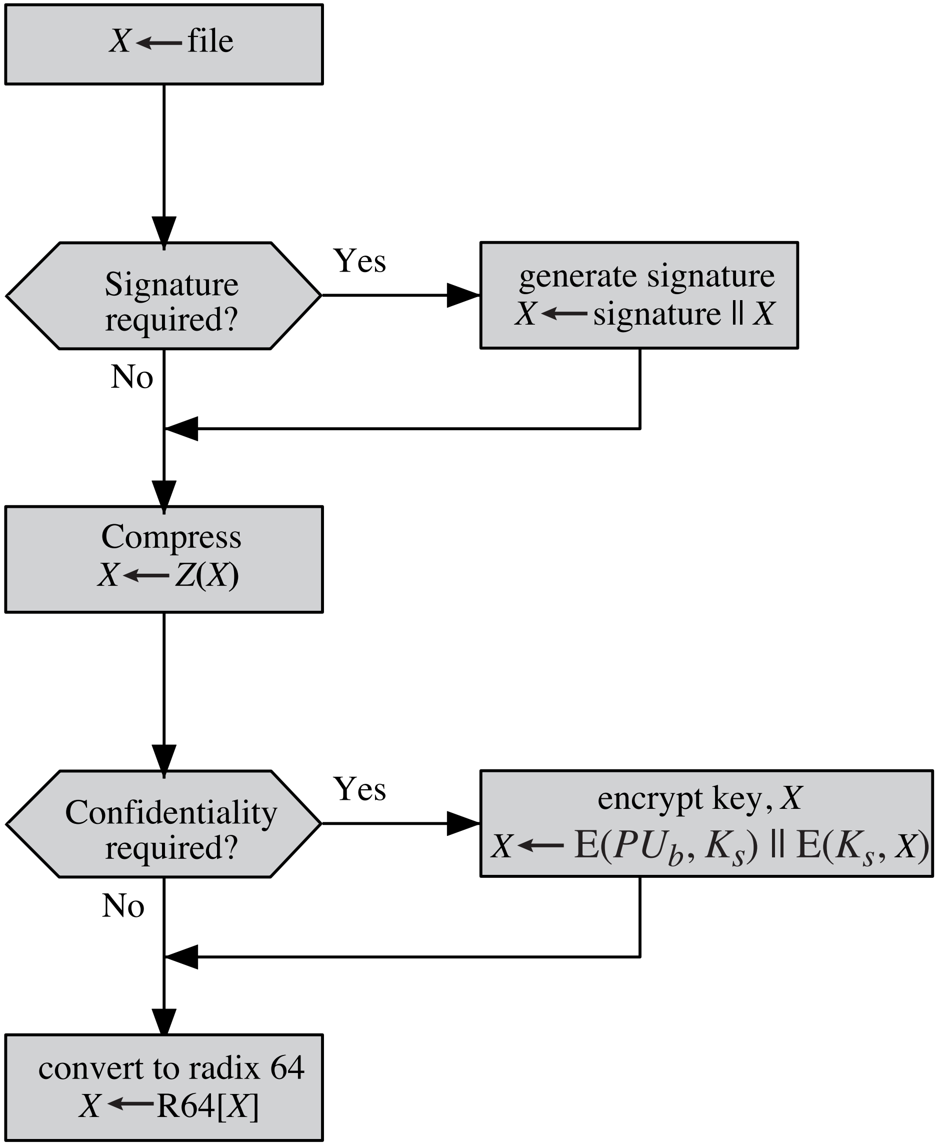




6-bit value	character encoding	6-bit value	character encoding
0	A	52	0
...	...	...	...
25	Z	61	9
26	a	62	+
...	...	63	/
51	z	(pad)	=

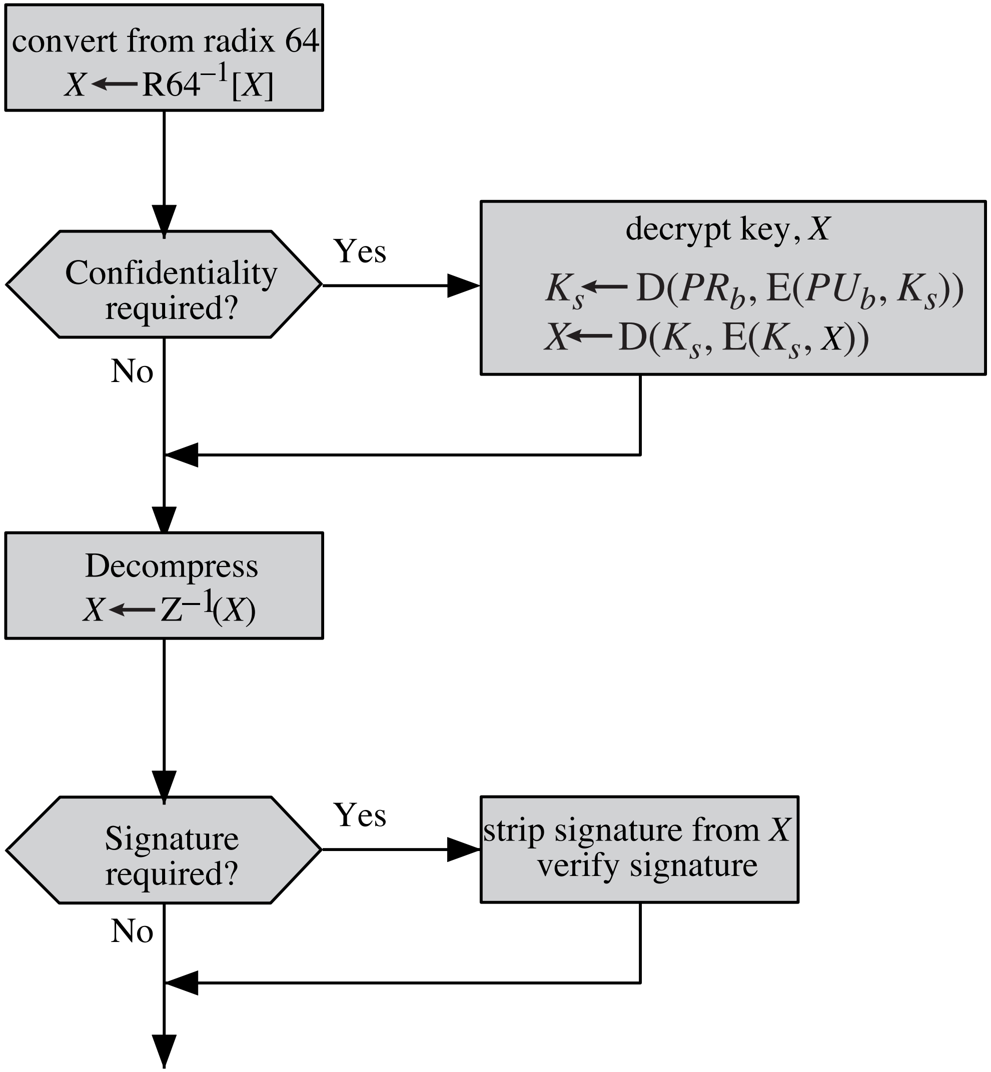
Processing of PGP messages

Generic transmission diagram (from A)



Processing of PGP messages

Generic reception diagram (to B)


convert from radix 64
$X \leftarrow \text{R64}^{-1}[X]$
Confidentiality required?
Yes
No
decrypt key, X
$K_s \leftarrow \text{D}(PR_b, \text{E}(PU_b, K_s))$
$X \leftarrow \text{D}(K_s, \text{E}(K_s, X))$
Decompress
$X \leftarrow \text{Z}^{-1}(X)$
Signature required?
Yes
No
strip signature from X
verify signature

PGP session keys

- need a session key for each message
  - of varying sizes: 56-bit DES, 128-bit CAST or IDEA, 168-bit Triple-DES
- generated using ANSI X12.17 mode
- uses random inputs taken from previous uses and from keystroke timing of user

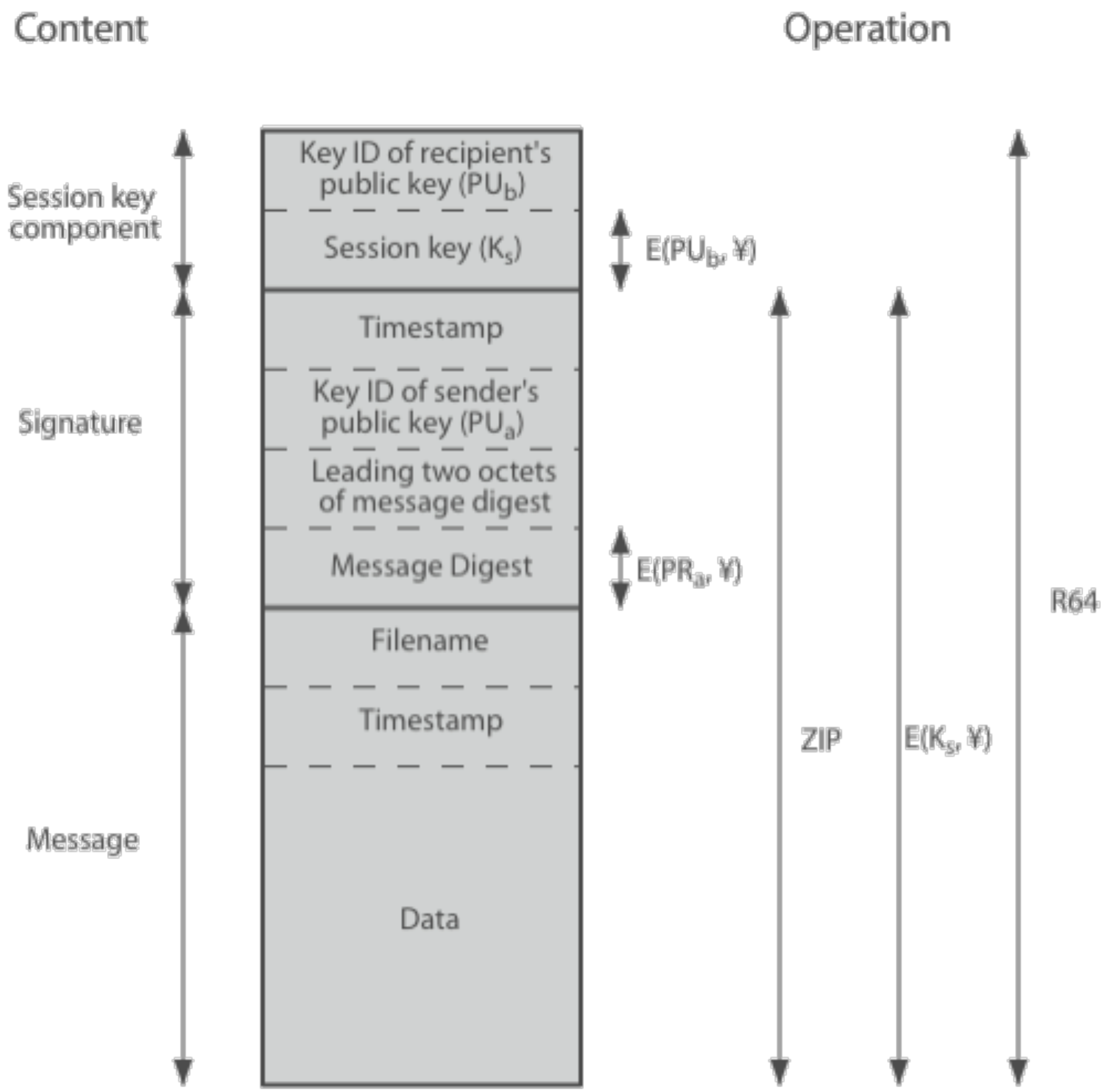
PGP random quantities

- PGP maintains a 256-byte buffer of random bits
- Each time PGP expects a keystroke from the user, it records
    - The time when it starts waiting (32 bits)
    - The time when the key was pressed (32 bits)
    - The value of the key stroke (8 bits)
- The recorded information is used to generate a key
- The generated key is used to encrypt the current value of the random-bit buffer

PGP public / private Keys

- many key pairs may be in use
  - need to identify which is used to encrypt session key in a message
  - could send full public-key with every message, but this is inefficient
- rather use a key identifier based on key
  - is least significant 64-bits of the key
  - will very likely be unique
- also use key ID in signatures

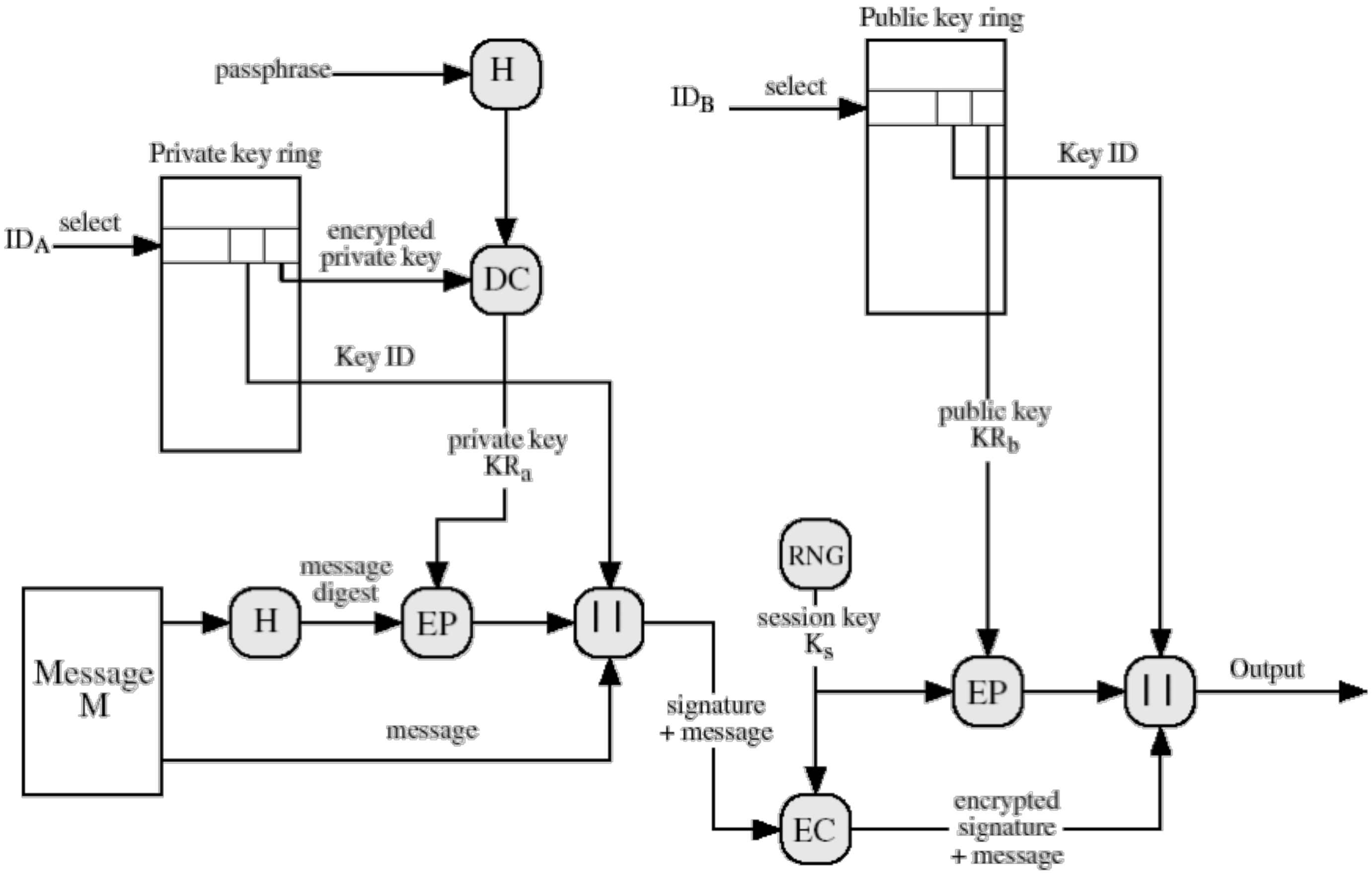
PGP message format



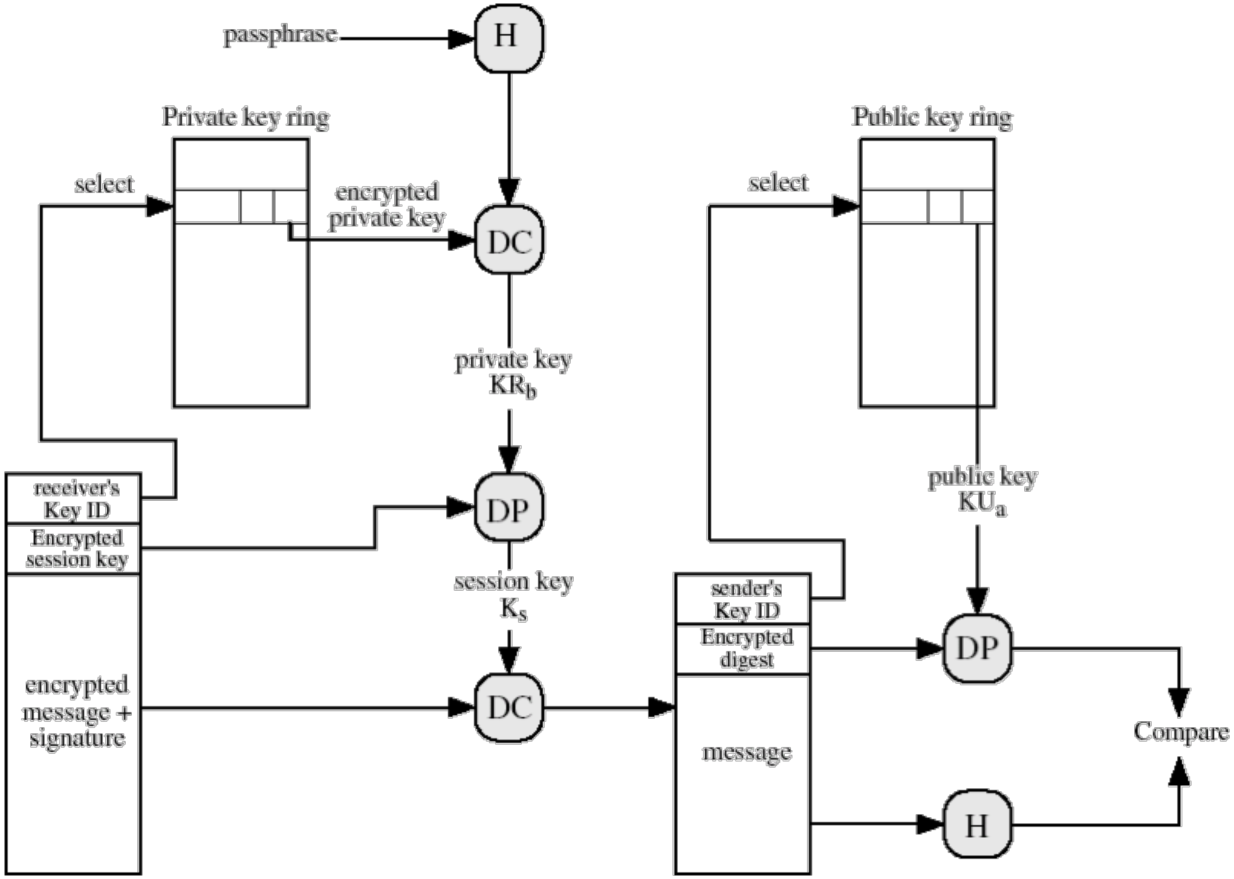
Example of signed message

```
---BEGIN PGP SIGNED MESSAGE---
Hash: SHA1

Bob:My husband is out of town
  tonight.Passionately yours, Alice

---BEGIN PGP SIGNATURE---
Version: PGP 5.0
Charset: noconv
yhHJRHhGJGhgg/12EpJ+lo8gE4vB3mqJhFEvZ
  P9t6n7G6m5Gw2
---END PGP SIGNATURE---

???(other data)???
---END PGP SIGNED MESSAGE---
```

PGP message transmission



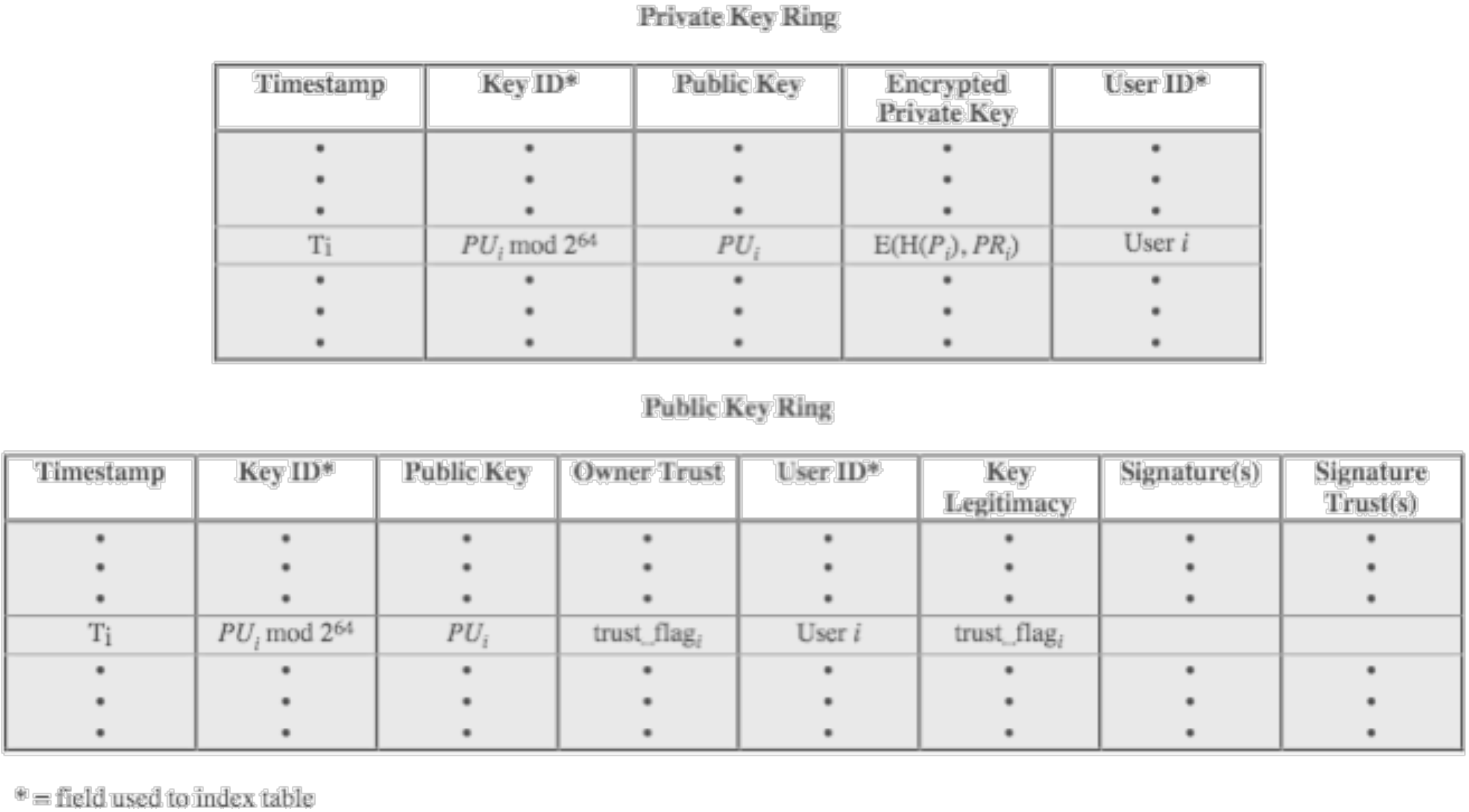
PGP message reception



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$_i$		
• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •

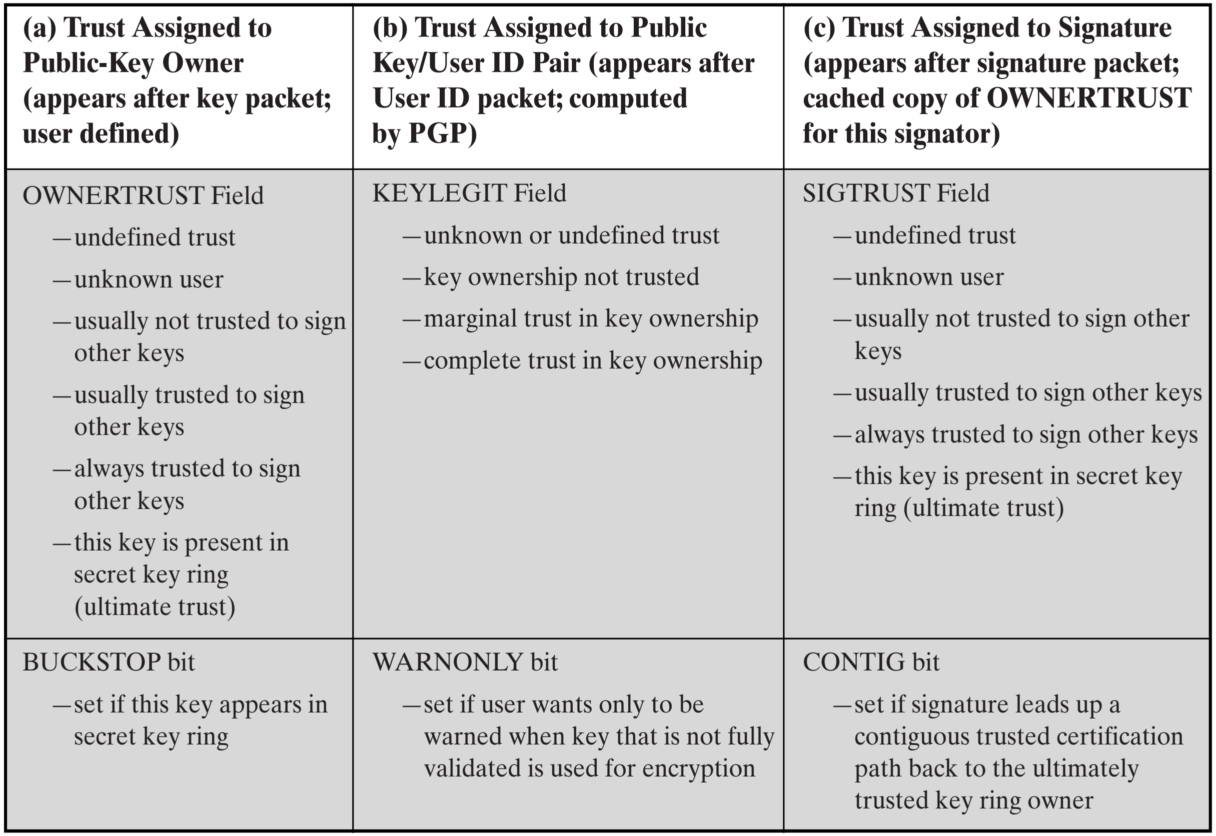
* = field used to index table

PGP key management

- rather than relying on certificate authorities
- in PGP every user is own CA
  - can sign keys for users they know directly
- forms a “web of trust”
  - trust keys have signed
  - trust keys others have signed if have a chain of signatures to them
- key ring includes trust indicators
- users can also revoke their keys

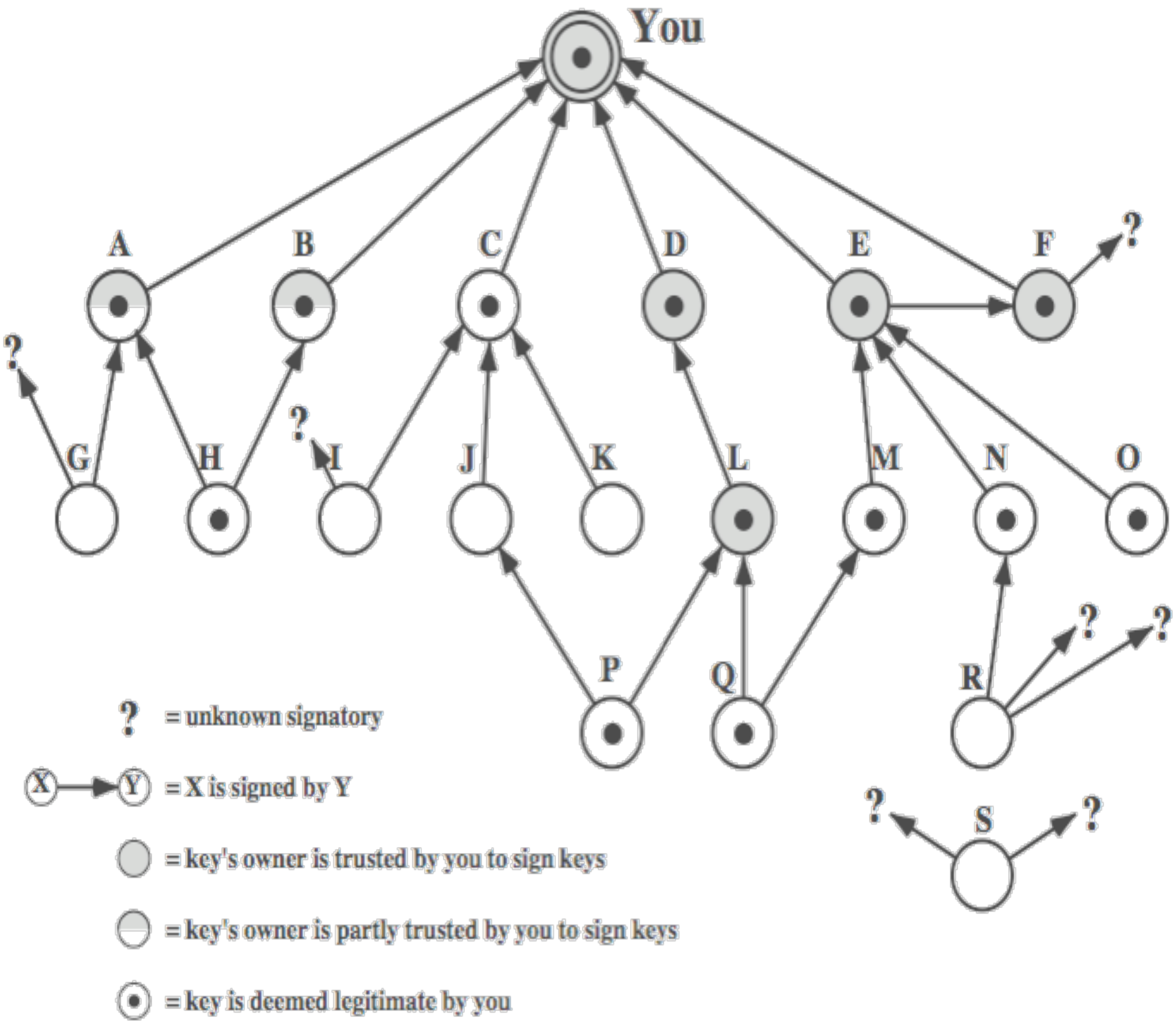
PGP signature trust

Contents of trust flag byte



(a) Trust Assigned to Public-Key Owner (appears after key packet; user defined)	(b) Trust Assigned to Public Key/User ID Pair (appears after User ID packet; computed by PGP)	(c) Trust Assigned to Signature (appears after signature packet; cached copy of OWNERTRUST for this signator)
OWNERTRUST Field —undefined trust —unknown user —usually not trusted to sign other keys —usually trusted to sign other keys —always trusted to sign other keys —this key is present in secret key ring (ultimate trust)	KEYLEGIT Field —unknown or undefined trust —key ownership not trusted —marginal trust in key ownership —complete trust in key ownership	SIGTRUST Field —undefined trust —unknown user —usually not trusted to sign other keys —usually trusted to sign other keys —always trusted to sign other keys —this key is present in secret key ring (ultimate trust)
BUCKSTOP bit —set if this key appears in secret key ring	WARNONLY bit —set if user wants only to be warned when key that is not fully validated is used for encryption	CONTIG bit —set if signature leads up a contiguous trusted certification path back to the ultimately trusted key ring owner

PGP trust model example


You
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
?
? = 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

PGP tools

- GnuPG
  https://www.gnupg.org
- PGP Desktop
  https://pgp.en.softonic.com/download
- OpenPGP
  https://www.openpgp.org

Secure MIME

- S/MIME is a security enhancement to the MIME standard
  - MIME: Multipurpose Internet Mail Extension
  - Based on technology from RSA Data Security

- Defined in
  - RFCs 3370, 3850, 3851, 3852
- Have S/MIME support in many email agents
  - MS Outlook
  - Mozilla
  - MacOS mail
  - …

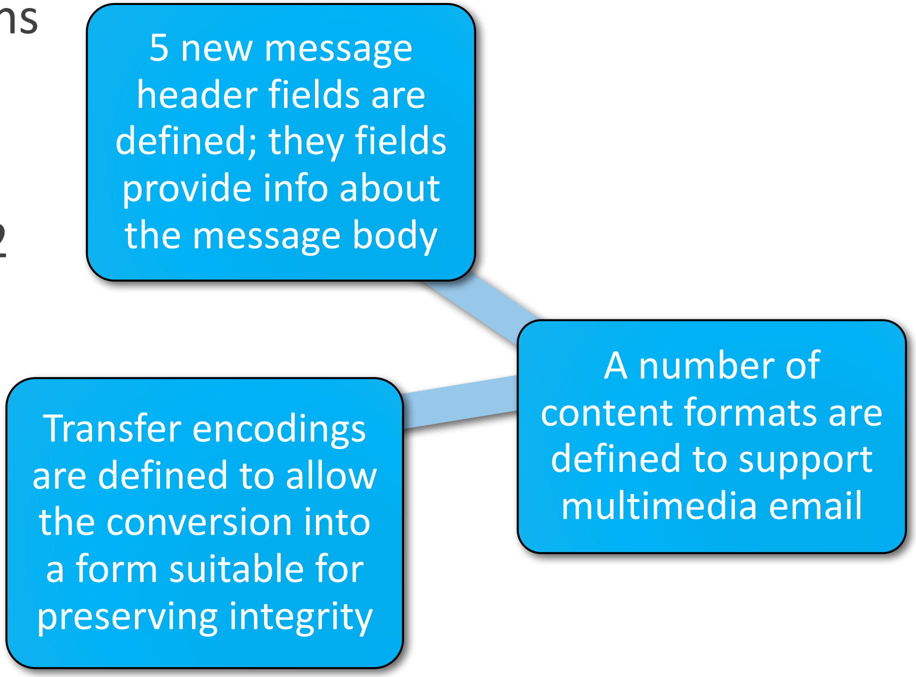
RFC 5322

- 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

About MIME

- Extension to RFC 5322
  - ○ intended to address some of the problems/limitations of the use of SMTP
  - ○ resolve in a manner compatible with RFC 5322 implementations
  - ○ The specification is provided in RFCs 2045 through 2049

- MIME spec. includes the following elements



5 new message header fields are defined; they fields provide info about the message body

A number of content formats are defined to support multimedia email

Transfer encodings are defined to allow the conversion into a form suitable for preserving integrity

MIME: header fields defined

MIME-Version

- Has the value 1.0; indicates that a message conforms to RFCs 2045 and 2046

Content-Type

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

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

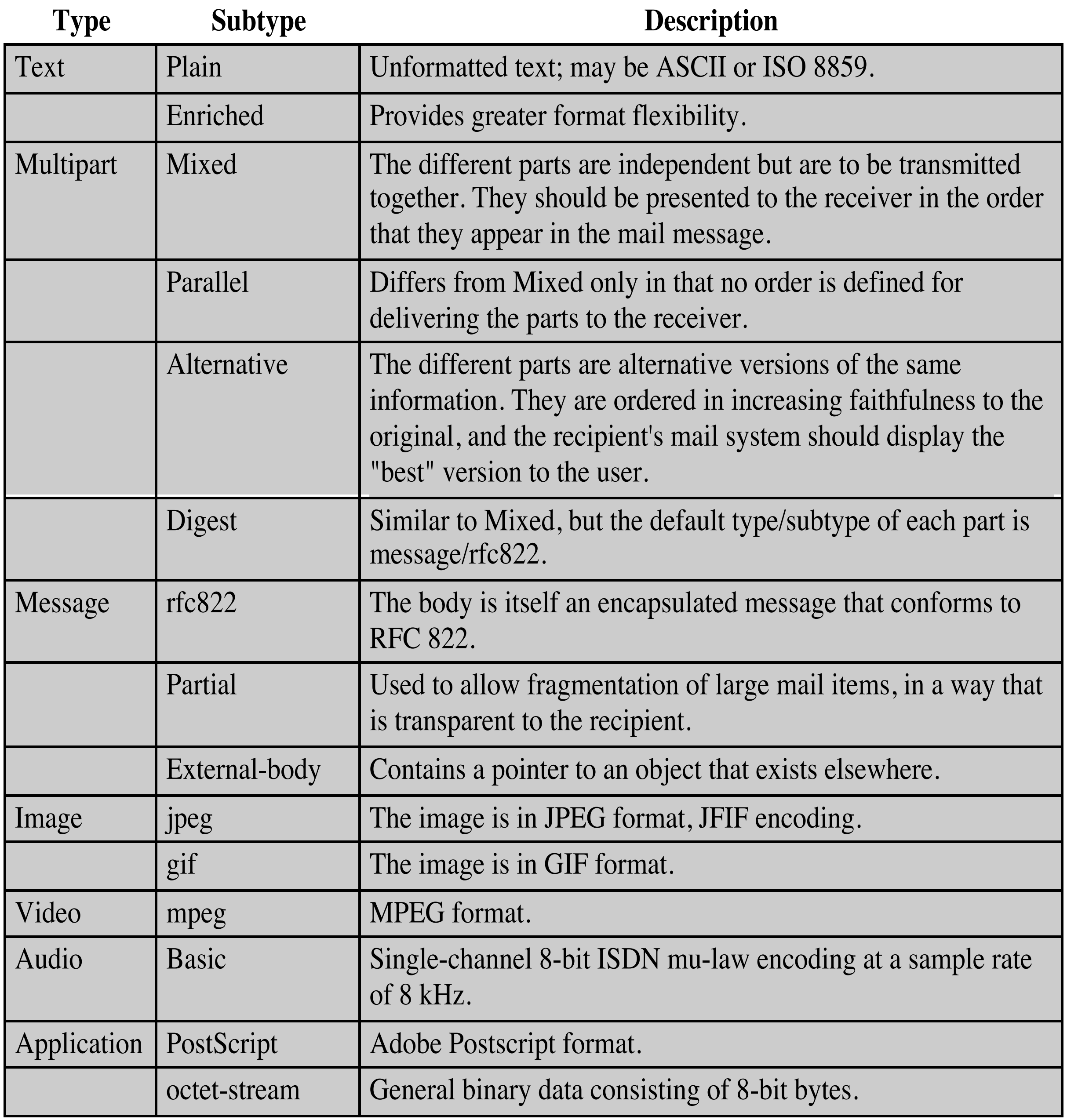
Content-ID

- Used to identify MIME entities uniquely in multiple contexts

Content-Description

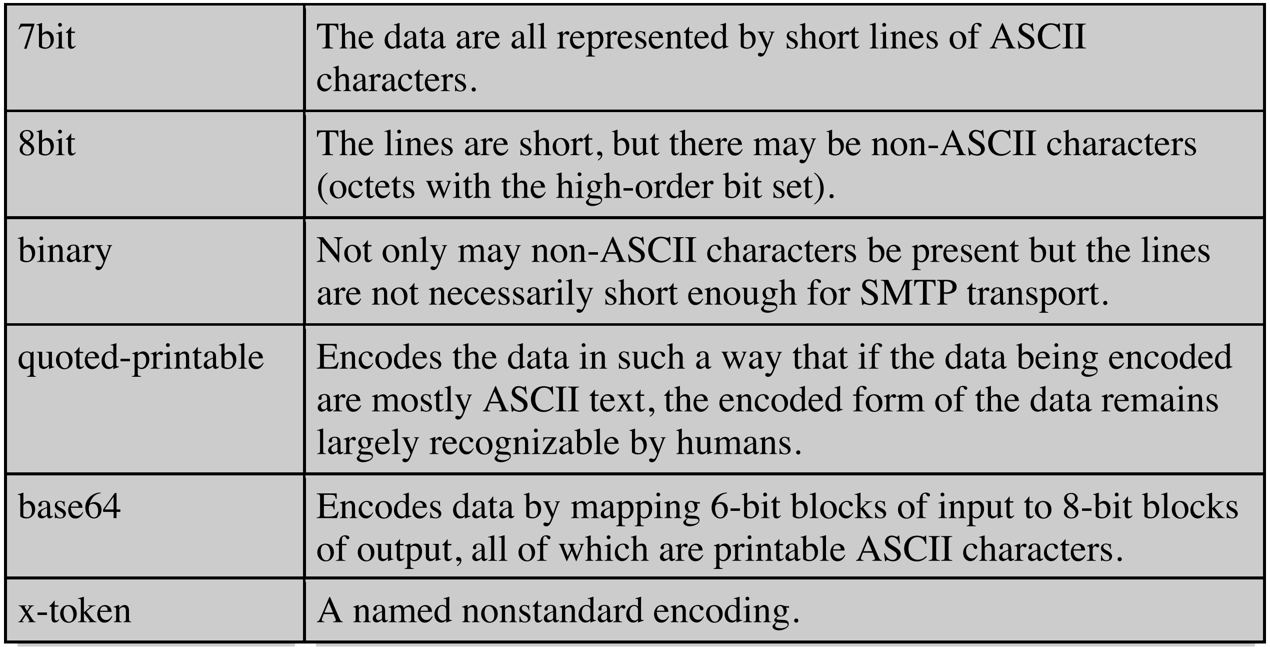
- Text description of an object with the body; useful if the object is not readable

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.

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.

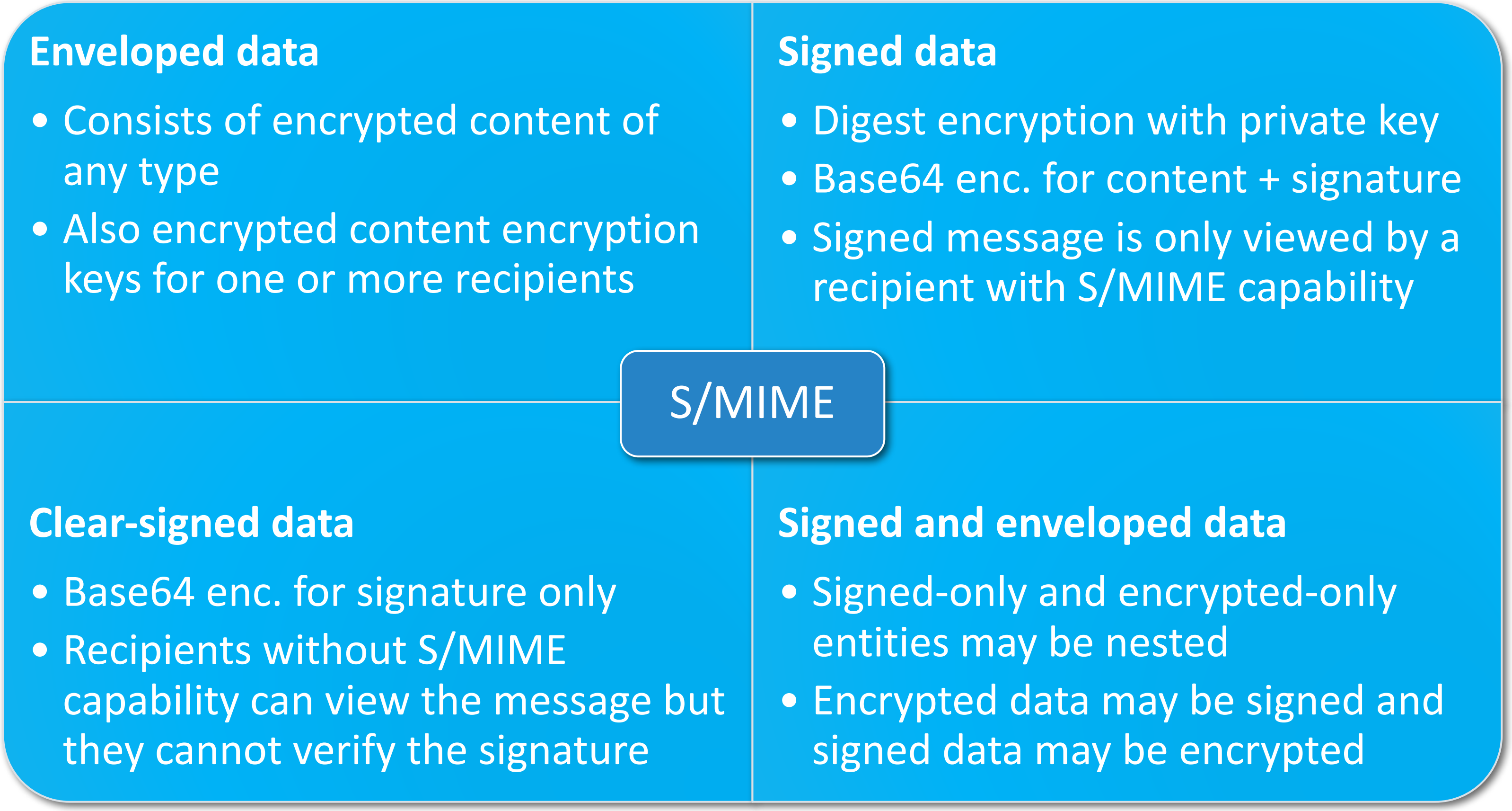
Example MIME message structure

```
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--
```

Native and canonical form

Native form	- The body to be transmitted is created in the system's native format; the native character set is used. - The body may be a text file or anything that corresponds to the local model for representing some form of information. - Fundamentally, the data is created in the "native" form that corresponds to the type specified by the media type.
Canonical form	- The body, incl. "out-of-band" information like record lengths or file attributes, is converted to a universal canonical form. - The specific media type of the body as well as its associated attributes dictate the nature of the canonical form that is used. - Conversion to the proper canonical form may involve character set conversion or other operations specific to a media type. - In case of character set conversion, care must be taken to understand the semantics of the media type.

S/MIME functionality



S/MIME

Enveloped data

- Consists of encrypted content of any type
- Also encrypted content encryption keys for one or more recipients

Signed data

- Digest encryption with private key
- Base64 enc. for content + signature
- Signed message is only viewed by a recipient with S/MIME capability

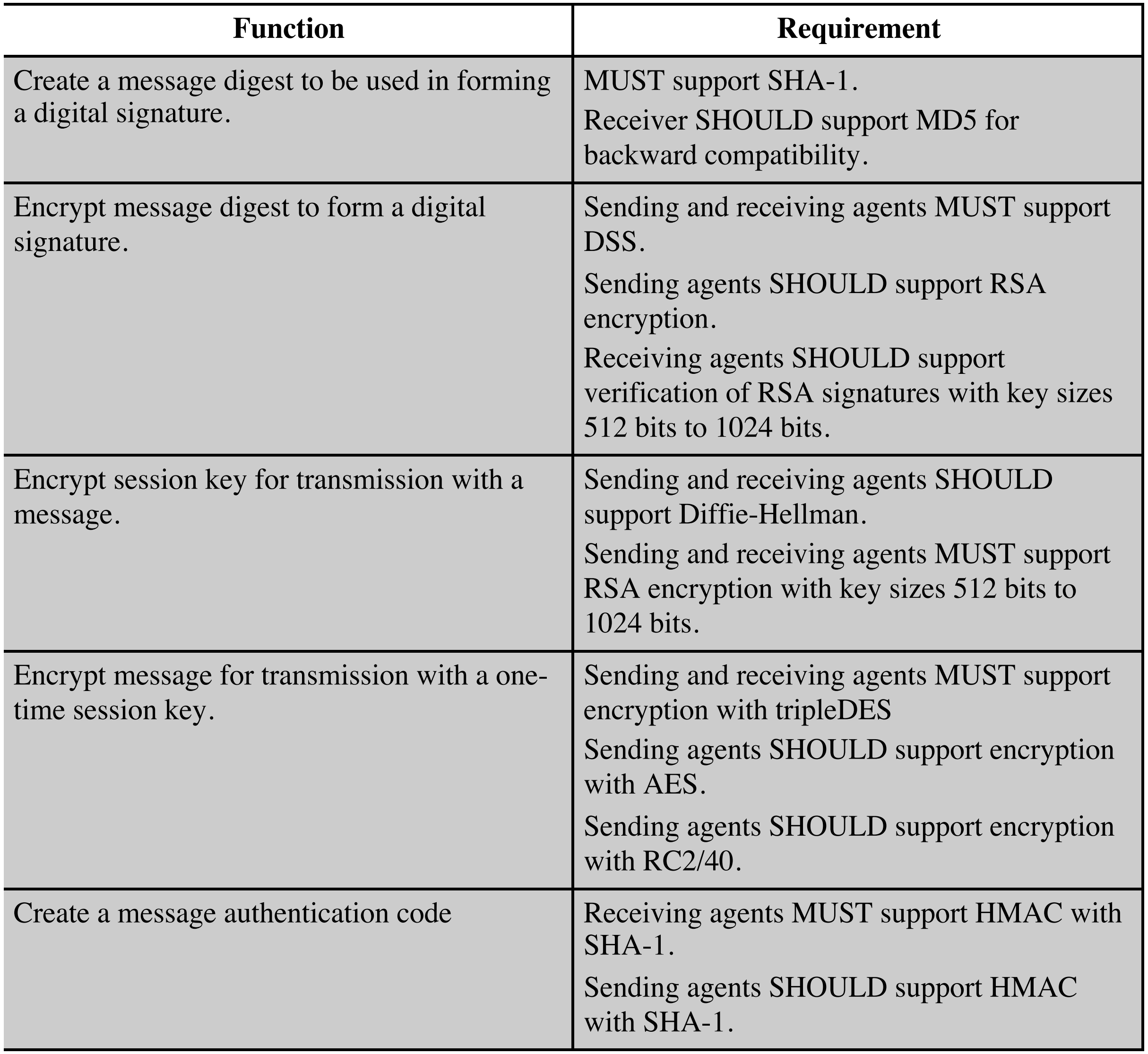
Clear-signed data

- Base64 enc. for signature only
- Recipients without S/MIME capability can view the message but they cannot verify the signature

Signed and enveloped data

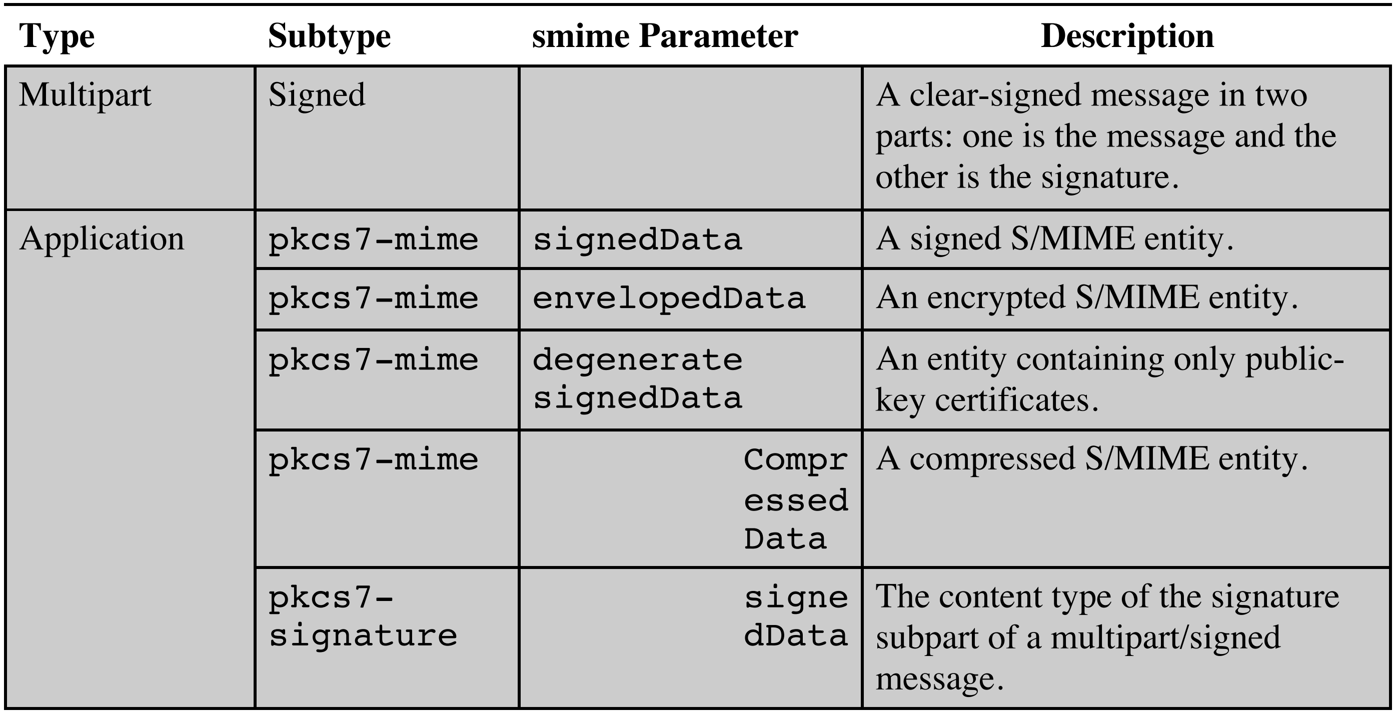
- Signed-only and encrypted-only entities may be nested
- Encrypted data may be signed and signed data may be encrypted

Crypto algorithms used in S/MIME



Function	Requirement
Create a message digest to be used in forming a digital signature.	MUST support SHA-1. Receiver SHOULD support MD5 for backward compatibility.
Encrypt message digest to form a digital signature.	Sending and receiving agents MUST support DSS. Sending agents SHOULD support RSA encryption. Receiving agents SHOULD support verification of RSA signatures with key sizes 512 bits to 1024 bits.
Encrypt session key for transmission with a message.	Sending and receiving agents SHOULD support Diffie-Hellman. Sending and receiving agents MUST support RSA encryption with key sizes 512 bits to 1024 bits.
Encrypt message for transmission with a one-time session key.	Sending and receiving agents MUST support encryption with tripleDES Sending agents SHOULD support encryption with AES. Sending agents SHOULD support encryption with RC2/40.
Create a message authentication code	Receiving agents MUST support HMAC with SHA-1. Sending agents SHOULD support HMAC with SHA-1.

S/MIME content types



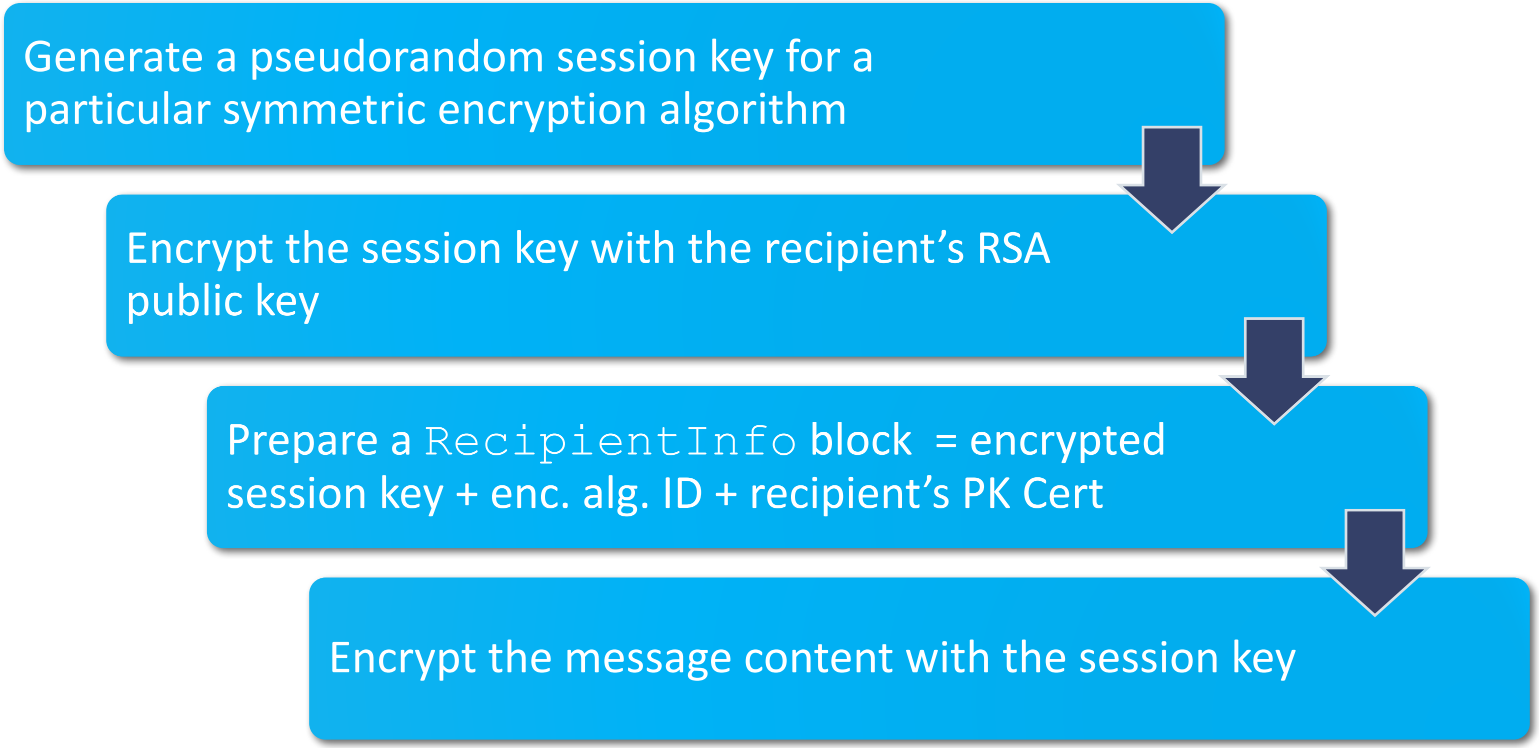
Type	Subtype	smime Parameter	Description
Multipart	Signed		A clear-signed message in two parts: one is the message and the other is the signature.
Application	`pkcs7-mime`	`signedData`	A signed S/MIME entity.
	`pkcs7-mime`	`envelopedData`	An encrypted S/MIME entity.
	`pkcs7-mime`	`degenerate signedData`	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, like alg. IDs and certificates, are processed by S/MIME to produce a PKCS object
  - A PKCS object is treated as message content and wrapped in MIME
- In all cases the message to be sent is converted to canonical form

Enveloped data

- The steps for preparing an EnvelopedData MIME are



Generate a pseudorandom session key for a particular symmetric encryption algorithm

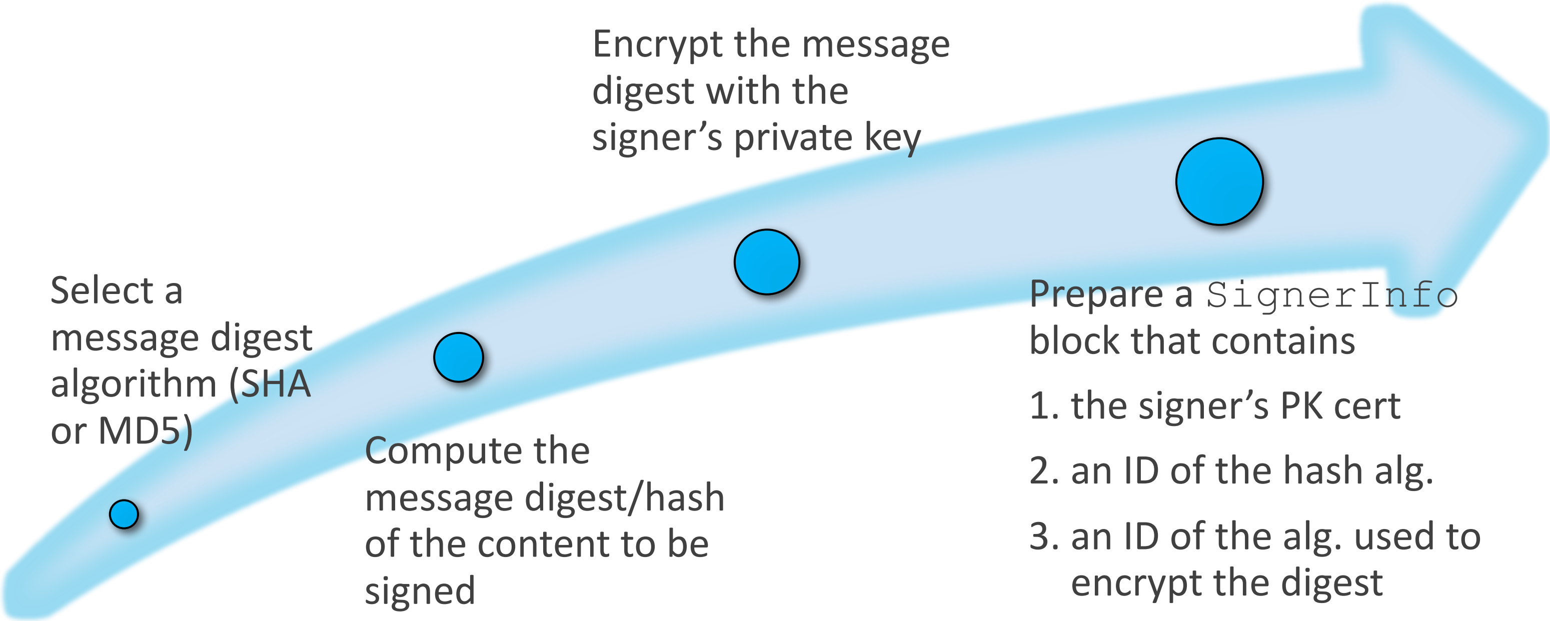
Encrypt the session key with the recipient’s RSA public key

Prepare a `RecipientInfo` block = encrypted session key + enc. alg. ID + recipient’s PK Cert

Encrypt the message content with the session key

Signed data

- The steps for preparing a SignedData MIME are



Select a message digest algorithm (SHA or MD5)

Compute the message digest/hash of the content to be signed

Encrypt the message digest with the signer’s private key

Prepare a `SignerInfo` block that contains

1. the signer’s PK cert
2. an ID of the hash alg.
3. an ID of the alg. used to encrypt the digest
4. the encrypted digest

Clear signing

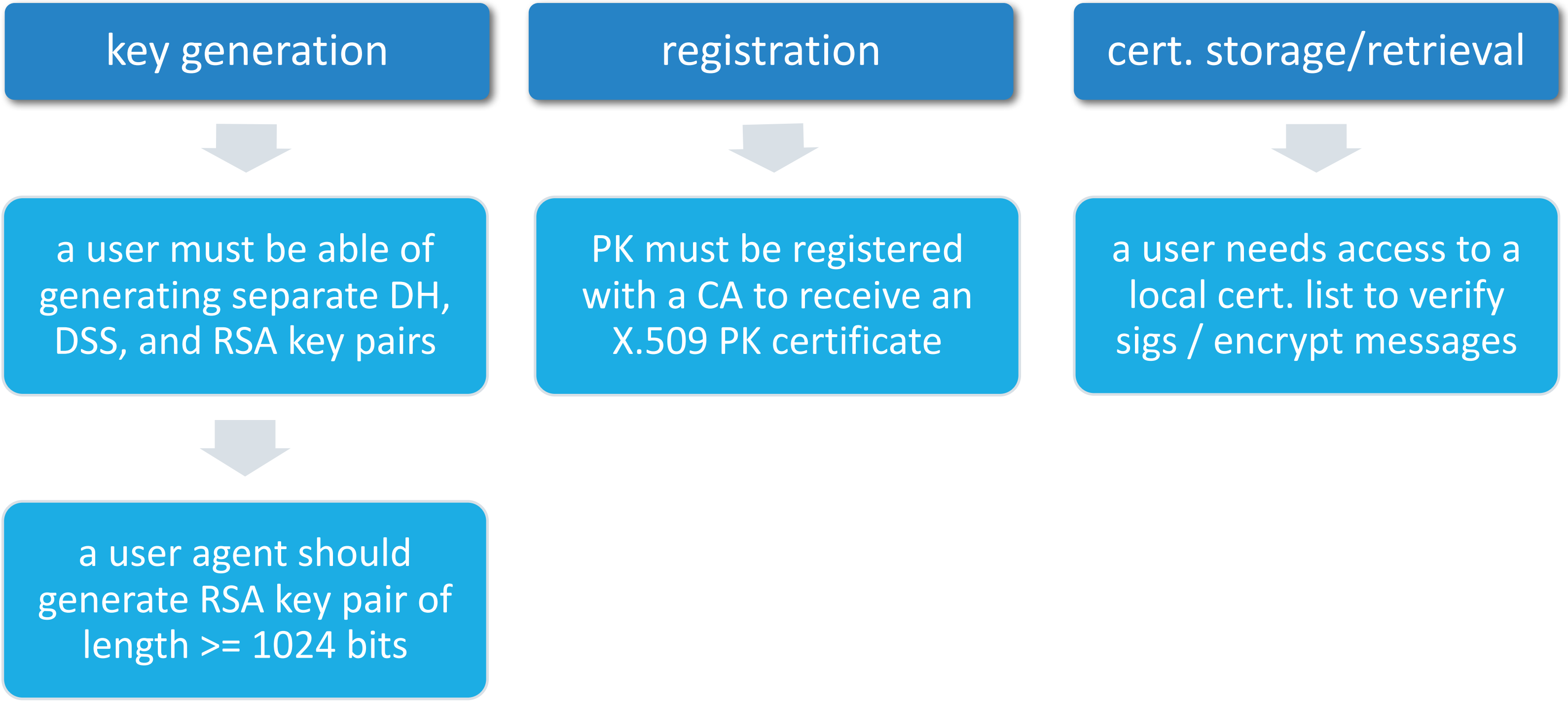
- Achieved using the multipart content type with a signed subtype
- This signing process does not involve transforming the message to be signed
- Recipients with MIME capability but not S/MIME capability are able to read the incoming message

S/MIME certificate processing

- S/MIME uses public-key certificates that conform to version 3 of X.509
- The key-management scheme used by S/MIME is ~ hybrid between an X.509 cert. hierarchy and PGP's web of trust
- S/MIME managers/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

- S/MIME users have key-management functions to perform



key generation

a user must be able of generating separate DH, DSS, and RSA key pairs

a user agent should generate RSA key pair of length >= 1024 bits

registration

PK must be registered with a CA to receive an X.509 PK certificate

cert. storage/retrieval

a user needs access to a local cert. list to verify sigs / encrypt messages

Enhanced security services

- Three enhanced security services have been proposed

Signed receipt

- Returning a signed receipt provides proof of delivery to the originator of a message and allows him to demonstrate 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

Κλάσεις πιστοποιητικών VeriSign

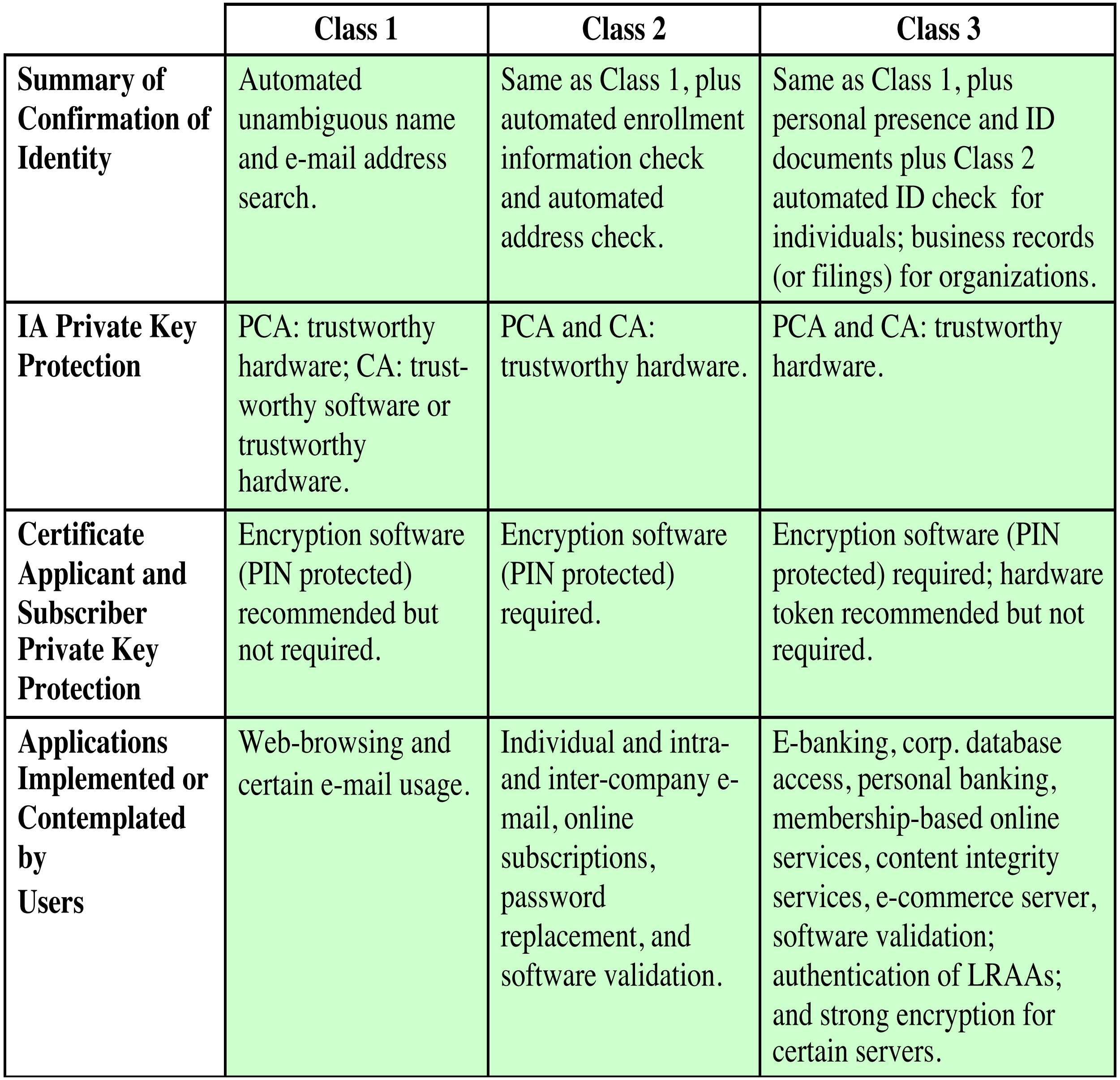
IA = Issuing Authority

CA = Certification Authority

PCA = VeriSign public primary certification authority

PIN = Personal Identification Number

LRAA = Local Registration Authority Administrator



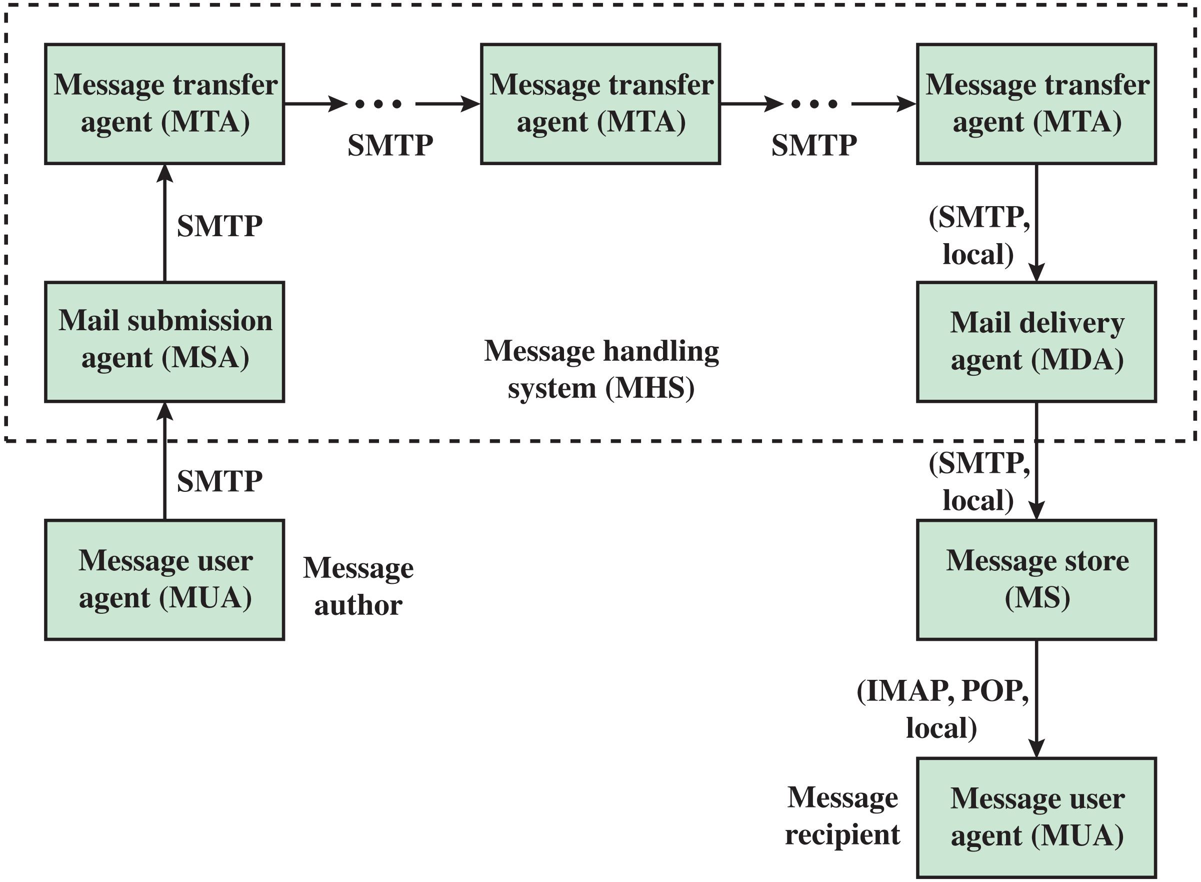
	Class 1	Class 2	Class 3
Summary of Confirmation of Identity	Automated unambiguous name and e-mail address search.	Same as Class 1, plus automated enrollment information check and automated address check.	Same as Class 1, plus personal presence and ID documents plus Class 2 automated ID check for individuals; business records (or filings) for organizations.
IA Private Key Protection	PCA: trustworthy hardware; CA: trustworthy software or trustworthy hardware.	PCA and CA: trustworthy hardware.	PCA and CA: trustworthy hardware.
Certificate Applicant and Subscriber Private Key Protection	Encryption software (PIN protected) recommended but not required.	Encryption software (PIN protected) required.	Encryption software (PIN protected) required; hardware token recommended but not required.
Applications Implemented or Contemplated by Users	Web-browsing and certain e-mail usage.	Individual and intra- and inter-company e-mail, online subscriptions, password replacement, and software validation.	E-banking, corp. database access, personal banking, membership-based online services, content integrity services, e-commerce server, software validation; authentication of LRAAs; and strong encryption for certain servers.

DomainKeys Identified Mail

- DKIM is a specification (in RFC 4871) for cryptographically signing e-mail
- Permits a signing domain to claim responsibility for a message in the mail stream
    - Recipients can verify the signature by querying the signer’s domain directly to retrieve the appropriate public key
    - Recipients can confirm that the message was attested to by a party in possession of the private key for the signing domain
- Has been widely adopted by e-mail providers and ISPs

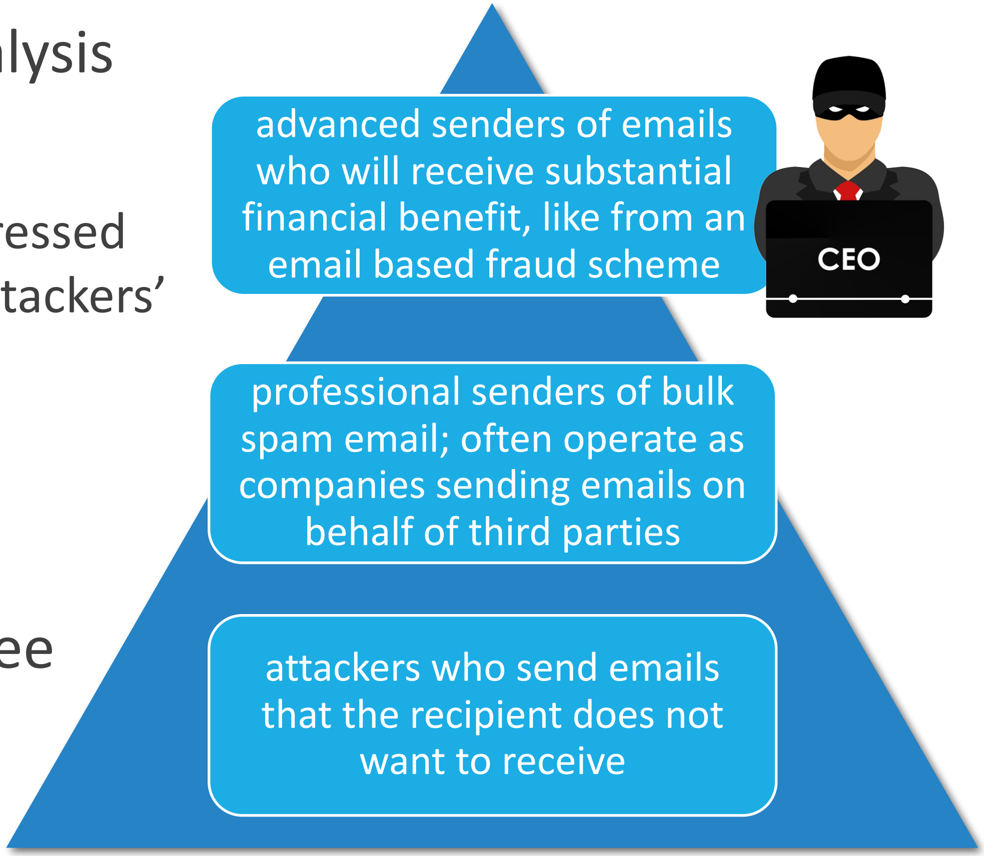
Function modules of DKIM

Standardized protocols used are also illustrated


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 threats

- RFC 4684 (threat analysis motivating DKIM)
  - Describes threats addressed by DKIM in terms of attackers'
    - capabilities
    - characteristics
    - location
- Characterized on three levels of threats



advanced senders of emails who will receive substantial financial benefit, like from an email based fraud scheme

professional senders of bulk spam email; often operate as companies sending emails on behalf of third parties

attackers who send emails that the recipient does not want to receive

E-mail threats: phishing stats

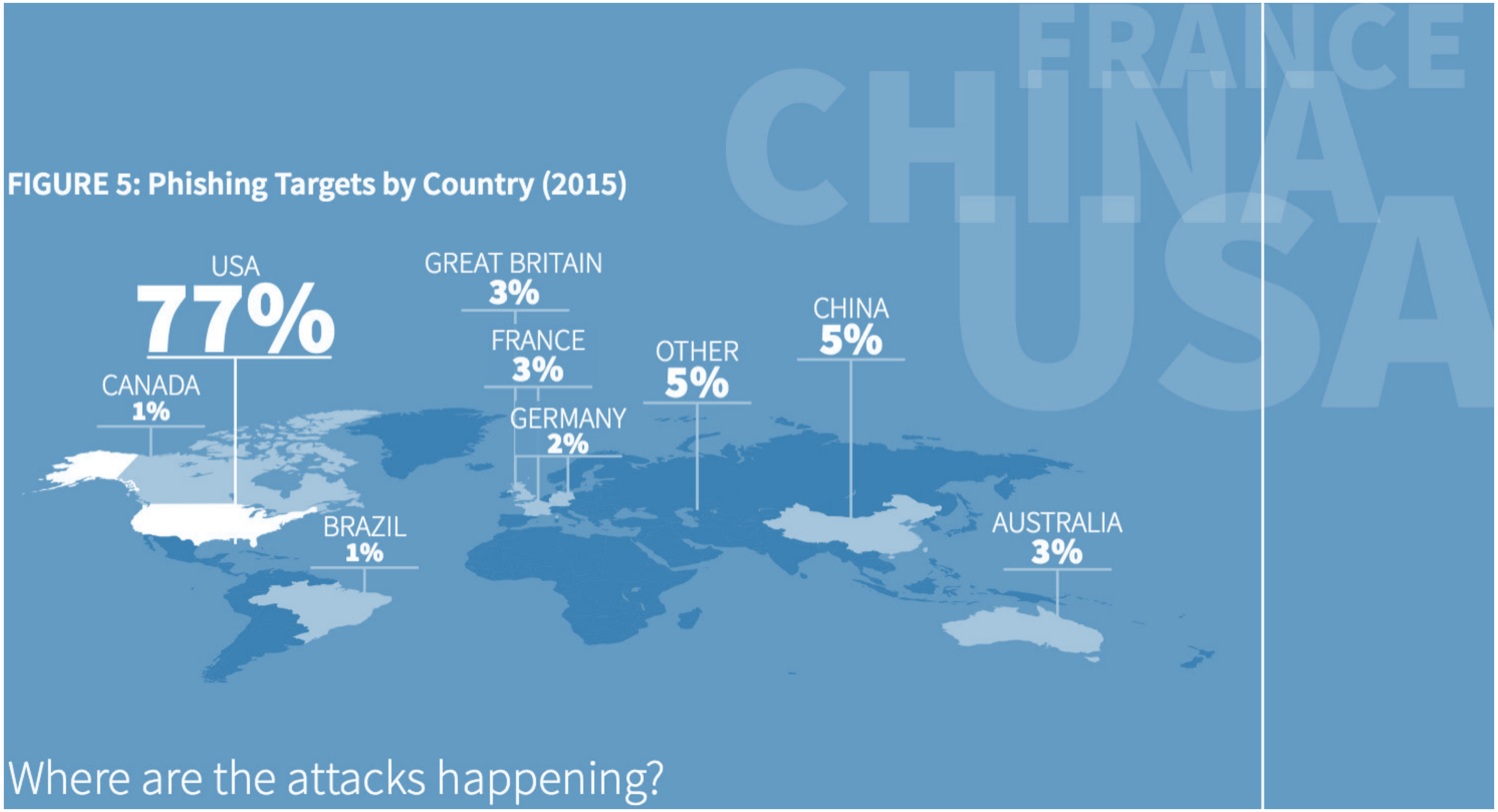
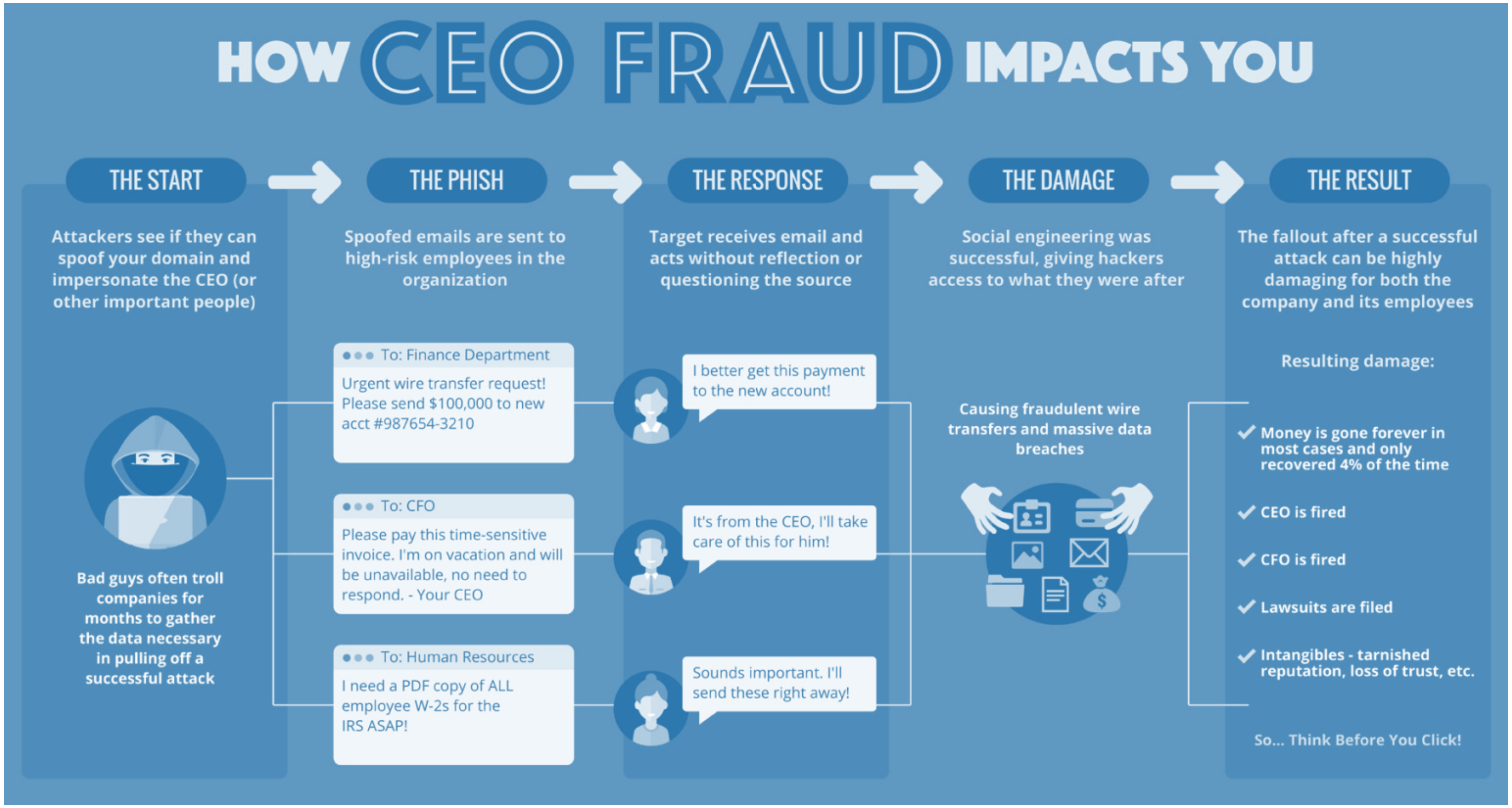


FIGURE 5: Phishing Targets by Country (2015)

Where are the attacks happening?

E-mail threats: CEO attack



HOW CEO FRAUD IMPACTS YOU

THE START

Attackers see if they can spoof your domain and impersonate the CEO (or other important people)

Bad guys often troll companies for months to gather the data necessary in pulling off a successful attack

THE PHISH

Spoofed emails are sent to high-risk employees in the organization

●●● To: Finance Department

Urgent wire transfer request! Please send $100,000 to new acct #987654-3210

●●● To: CFO

Please pay this time-sensitive invoice. I'm on vacation and will be unavailable, no need to respond. - Your CEO

●●● To: Human Resources

I need a PDF copy of ALL employee W-2s for the IRS ASAP!

THE RESPONSE

Target receives email and acts without reflection or questioning the source

I better get this payment to the new account!

It's from the CEO, I'll take care of this for him!

Sounds important. I'll send these right away!

THE DAMAGE

Social engineering was successful, giving hackers access to what they were after

Causing fraudulent wire transfers and massive data breaches

THE RESULT

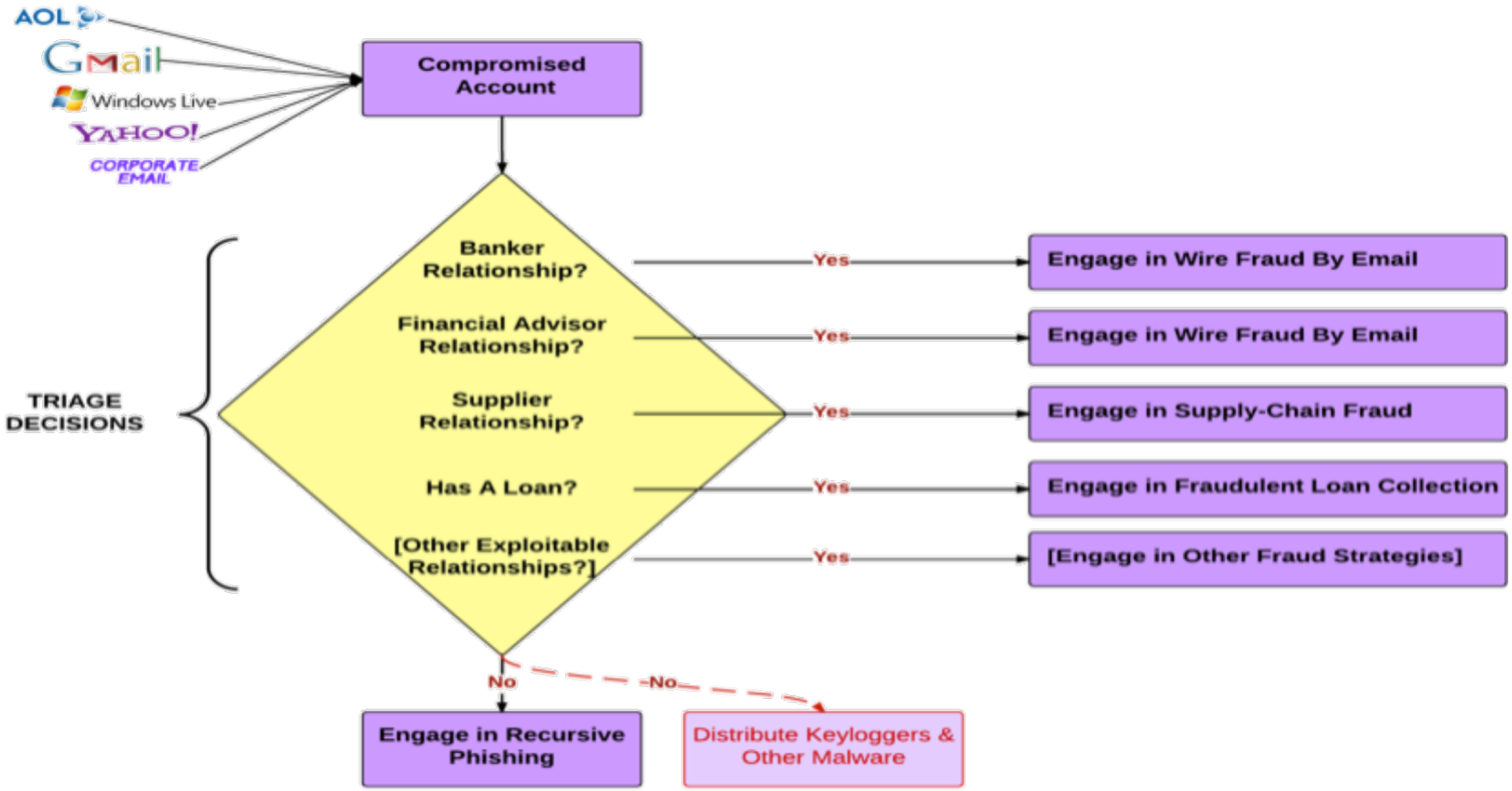
The fallout after a successful attack can be highly damaging for both the company and its employees

Resulting damage:

- ✓ Money is gone forever in most cases and only recovered 4% of the time
- ✓ CEO is fired
- ✓ CFO is fired
- ✓ Lawsuits are filed
- ✓ Intangibles - tarnished reputation, loss of trust, etc.

So... Think Before You Click!

Man-in-the-email (MITE)



AOL
Gmail
Windows Live
YAHOO!
CORPORATE EMAIL
Compromised Account
TRIAGE DECISIONS
Banker Relationship?
Financial Advisor Relationship?
Supplier Relationship?
Has A Loan?
[Other Exploitable Relationships?]
Yes
Yes
Yes
Yes
Yes
Engage in Wire Fraud By Email
Engage in Wire Fraud By Email
Engage in Supply-Chain Fraud
Engage in Fraudulent Loan Collection
[Engage in Other Fraud Strategies]
No
No
Engage in Recursive Phishing
Distribute Keyloggers & Other Malware

E-mail threats: losses (2016)

US ONLY

- number of victims reported: 17.642
- more than $2.3bn losses
- 270% increase since 2015

GLOBALLY

- number of victims reported: 22.143
- more than $3.1bn losses
- 1.300% increase since 2015

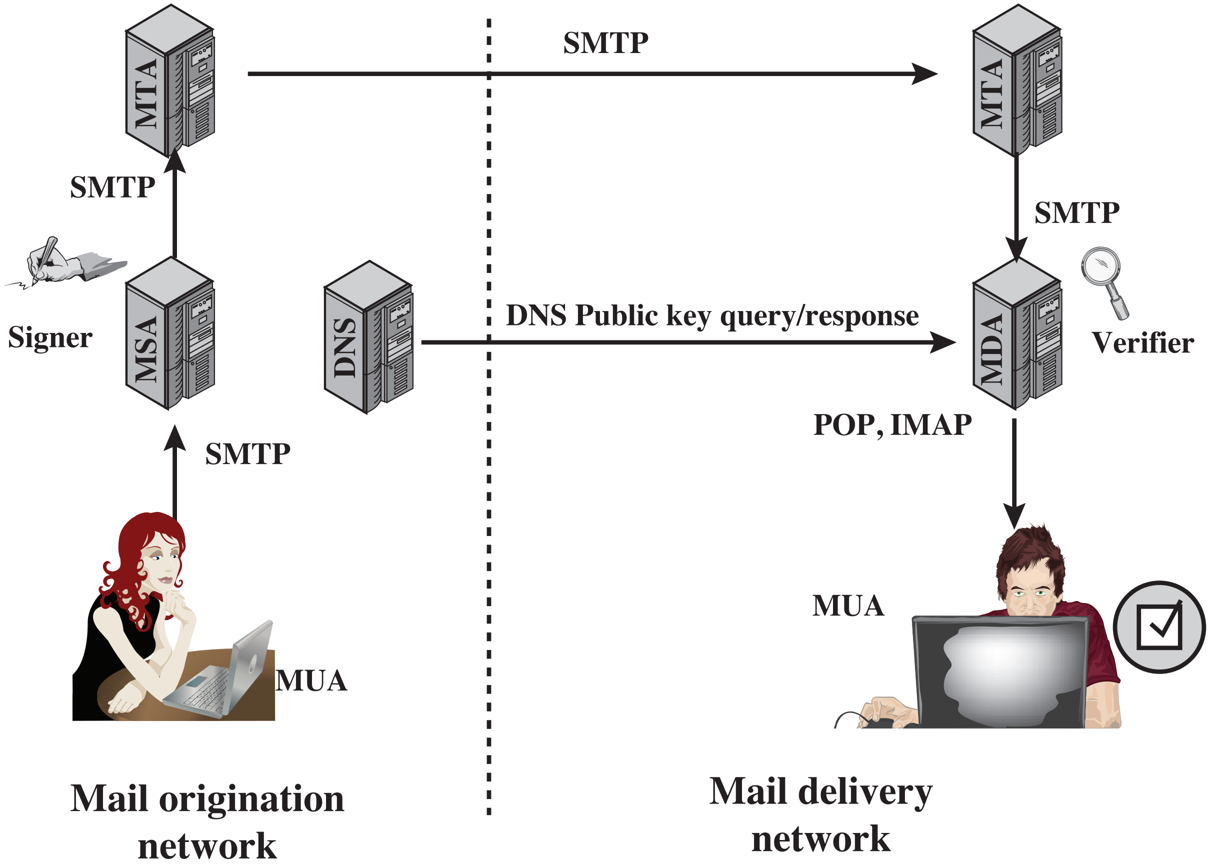
Source: https://www.ic3.gov/media/2017/170504.aspx#fn3

Simple example of DKIM deployment

Transparent to user

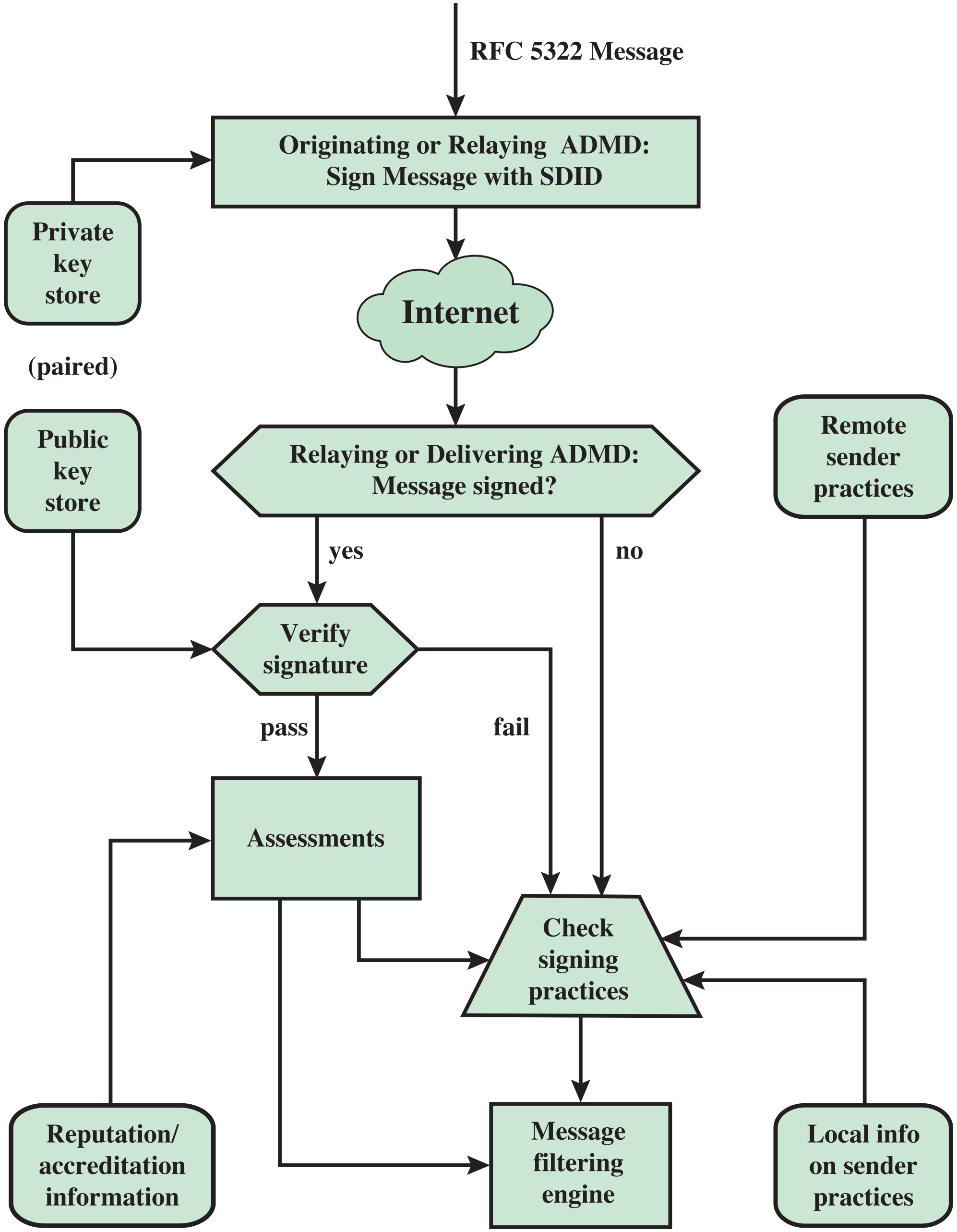
- MSA sign
- MDA verify

for pragmatic reasons

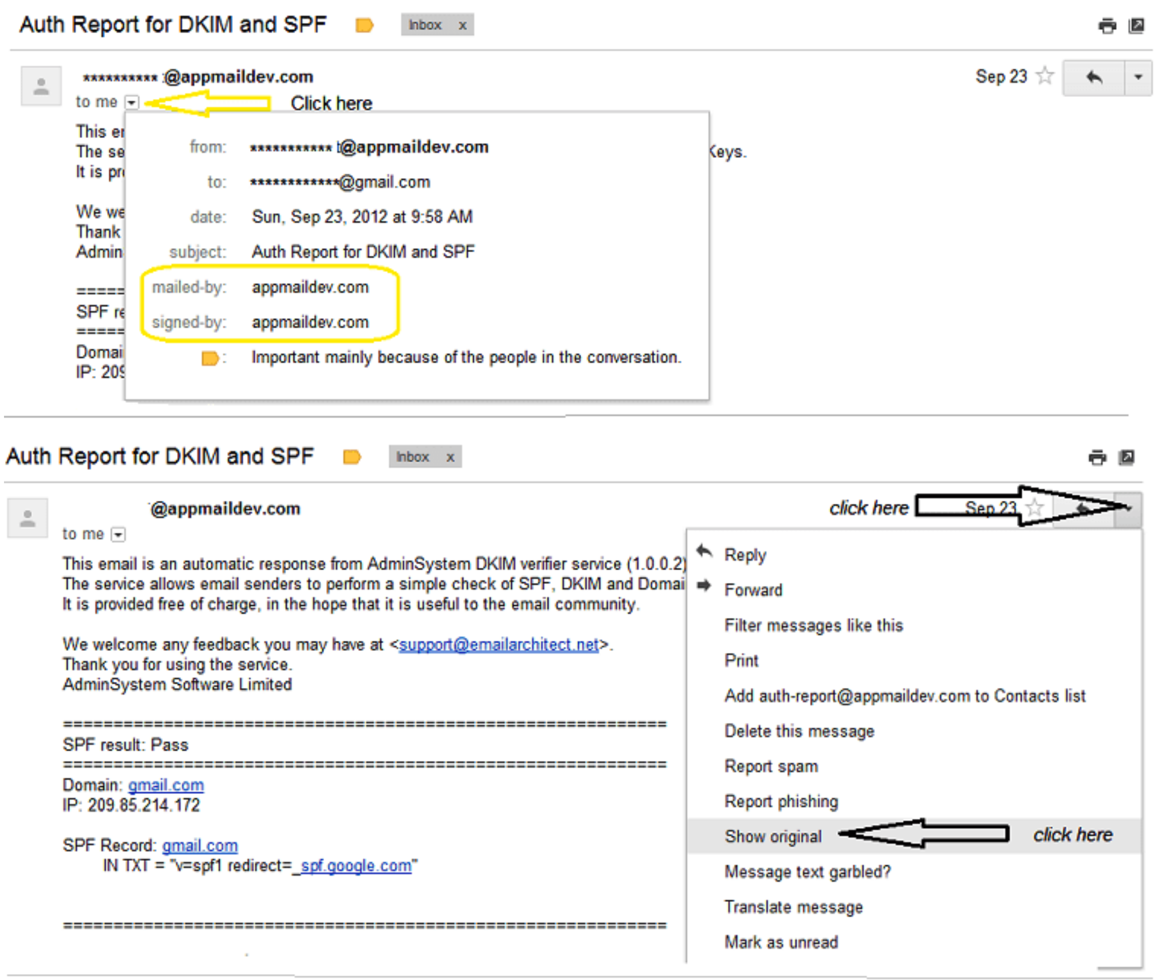


DNS = domain name system
MDA = mail delivery agent
MSA = mail submission agent
MTA = message transfer agent
MUA = message user agent

DKIM functional flow



Testing DKIM with Gmail



Auth Report for DKIM and SPF

********** @appmaildev.com

to me

Sep 23

from: *********** @appmaildev.com

to: ************@gmail.com

date: Sun, Sep 23, 2012 at 9:58 AM

subject: Auth Report for DKIM and SPF

mailed-by: appmaildev.com

signed-by: appmaildev.com

Important mainly because of the people in the conversation.

Auth Report for DKIM and SPF

@appmaildev.com

to me

Sep 23

This email is an automatic response from AdminSystem DKIM verifier service (1.0.0.2)
The service allows email senders to perform a simple check of SPF, DKIM and Domai
It is provided free of charge, in the hope that it is useful to the email community.

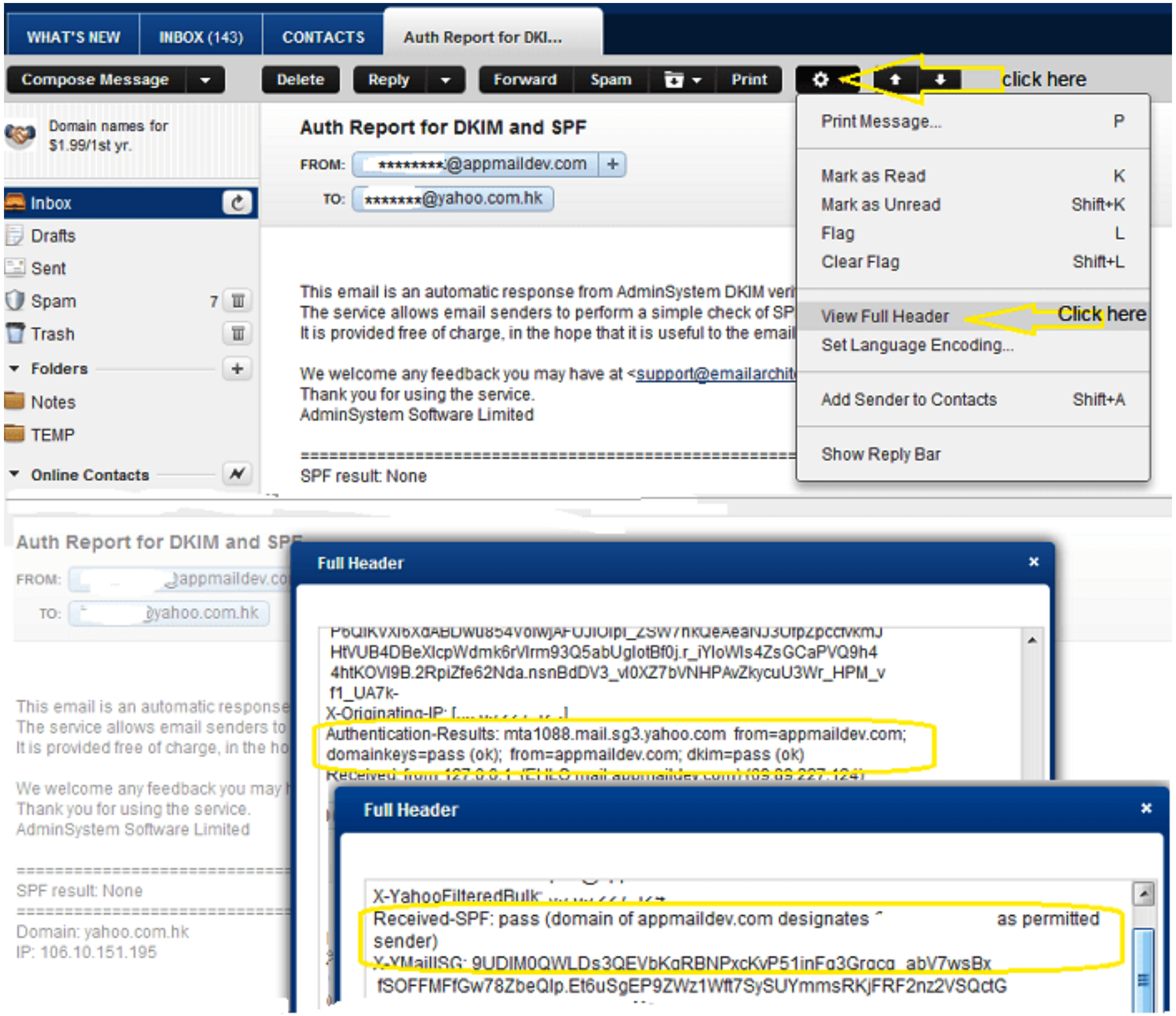
We welcome any feedback you may have at <support@emailarchitect.net>.
Thank you for using the service.
AdminSystem Software Limited

```
============================================================
SPF result: Pass
============================================================
Domain: gmail.com
IP: 209.85.214.172

SPF Record: gmail.com
      IN TXT = "v=spf1 redirect=_spf.google.com"


============================================================
```

You will see the following headers: dkim=pass

```
Received: from mail.appmaildev.com (mail.emailarchitect.com. [69.89.227.124])
        by mx.google.com with ESMTP id fz4si17691265qab.11.2012.09.22.18.56.01;
        Sat, 22 Sep 2012 18:56:02 -0700 (PDT)
Received-SPF: pass (google.com: domain of au          @appmaildev.com designates                as permit
Authentication-Results: mx.google.com; spf=pass (google.com: domain of           :@appmaildev.com desig
      @appmaildev.com; dkim=pass header.i=@appmaildev.com
DKIM-Signature: v=1; a=rsa-sha1; c=relaxed/relaxed;
  s=s1024;d=appmaildev.com;
```

Testing DKIM with Yahoo



Auth Report for DKIM and SPF

FROM: *********@appmaildev.com

TO: *******@yahoo.com.hk

This email is an automatic response from AdminSystem DKIM veri
The service allows email senders to perform a simple check of SP
It is provided free of charge, in the hope that it is useful to the email

We welcome any feedback you may have at <support@emailarchi
Thank you for using the service.
AdminSystem Software Limited

==
SPF result: None

Auth Report for DKIM and SPF

FROM: appmaildev.co

TO: yahoo.com.hk

This email is an automatic response
The service allows email senders to
It is provided free of charge, in the ho

We welcome any feedback you may h
Thank you for using the service.
AdminSystem Software Limited

==============================
SPF result: None
==============================

Domain: yahoo.com.hk
IP: 106.10.151.195

```
P6QIKVXI6XdABDWu854VolWjAFUJIOIpl_ZSW7nkQeAeaNJ3UfpZpccfvkmJ
 HtVUB4DBeXlcpWdmk6rVlrm93Q5abUglotBf0j.r_iYloWls4ZsGCaPVQ9h4
 4htKOVl9B.2RpiZfe62Nda.nsnBdDV3_vl0XZ7bVNHPAvZkycuU3Wr_HPM_v
 f1_UA7k-
X-Originating-IP: [
Authentication-Results: mta1088.mail.sg3.yahoo.com  from=appmaildev.com;
domainkeys=pass (ok);  from=appmaildev.com; dkim=pass (ok)
Received: from 127.0.0.1 (EHLO mail.appmaildev.com) (
```

```
X-YahooFilteredBulk:
Received-SPF: pass (domain of appmaildev.com designates                as permitted
sender)
X-YMailISG: 9UDIM0QWLDs3QEVbKgRBNPxcKvP51inFg3Grgcg_abV7wsBx
 fSOFFMFfGw78ZbeQlp.Et6uSgEP9ZWz1Wft7SySUYmmsRKjFRF2nz2VSQctG
```

DKIM advantages

- Authentication and authorization help receiving systems to know, with absolute assurance, who the real users are
- For very large mailing lists, the email blocking limits are often raised
  - If no DKIM authentication is used, emails are likely to get blocked by major ISPs like Gmail, Yahoo, and Hotmail
- Potentially less stringent SPAM filtering
  - If marketing messages are sent, email firewalls can be harsh

Προτεινόμενη βιβλιογραφία

- W. Stallings
  Cryptography and Network Security: Principles & Practice
  7th Ed., Prentice Hall, 2017