Electronic Mail Security

Raj Jain
Washington University in Saint Louis
Saint Louis, MO 63130
Jain@cse.wustl.edu

Audio/Video recordings of this lecture are available at:

http://www.cse.wustl.edu/~jain/cse571-11/

Overview

1. Pretty Good Privacy (PGP)
2. S/MIME
3. DomainKeys Identified Mail (DKIM)

These slides are based partly on Lawrie Brown's slides supplied with William Stallings's book "Cryptography and Network Security: Principles and Practice," 5th Ed, 2011.

Email Security Enhancements

1. Confidentiality: Protection from disclosure

- Authentication: Of sender of message
- Message integrity: Protection from modification
- Non-repudiation of origin: Protection from denial by sender

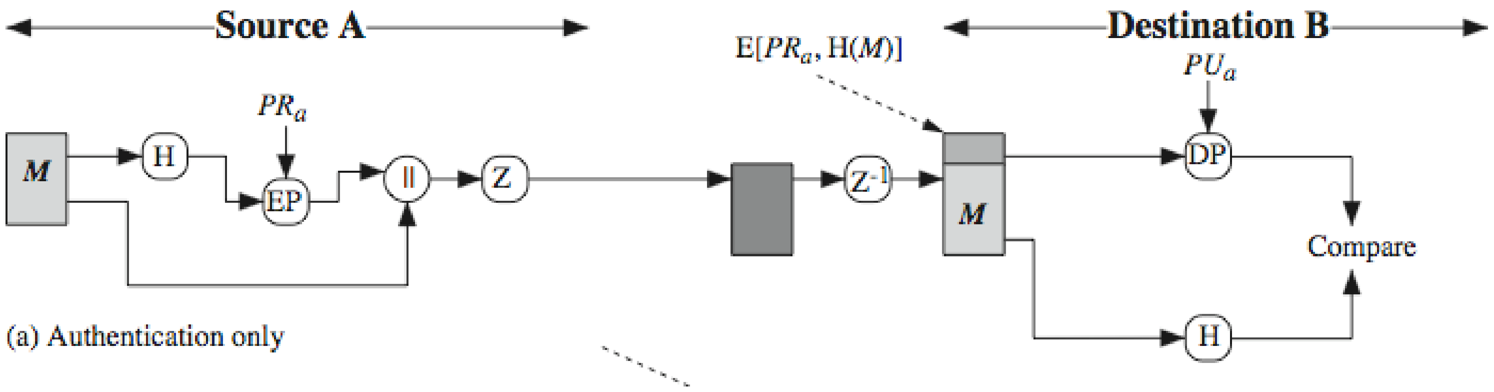
Pretty Good Privacy (PGP)

- Widely used de facto secure email
- Developed by Phil Zimmermann
- Selected best available crypto algorithms
- Integrated into a single program
- On Unix, PC, Macintosh and other systems
- Originally free, now also have commercial versions available
- Published as an OCRable book from MIT Press to allow export
- OpenPGP standard from IETF

Ref: http://en.wikipedia.org/wiki/Pretty_Good_Privacy

PGP Operation – Authentication

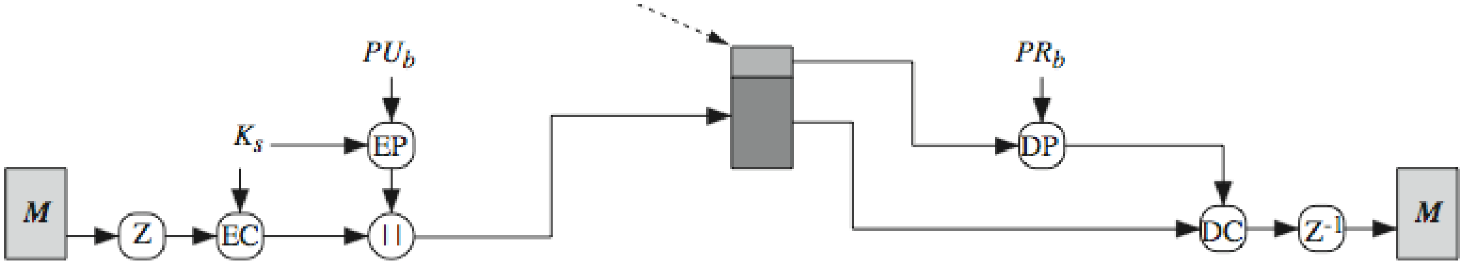
1. Sender creates message
2. Make SHA-1 160-bit hash of message
3. Attached RSA signed hash to message
4. Receiver decrypts & recovers hash code
5. Receiver verifies received message hash



(a) Authentication only

PGP Operation – Confidentiality

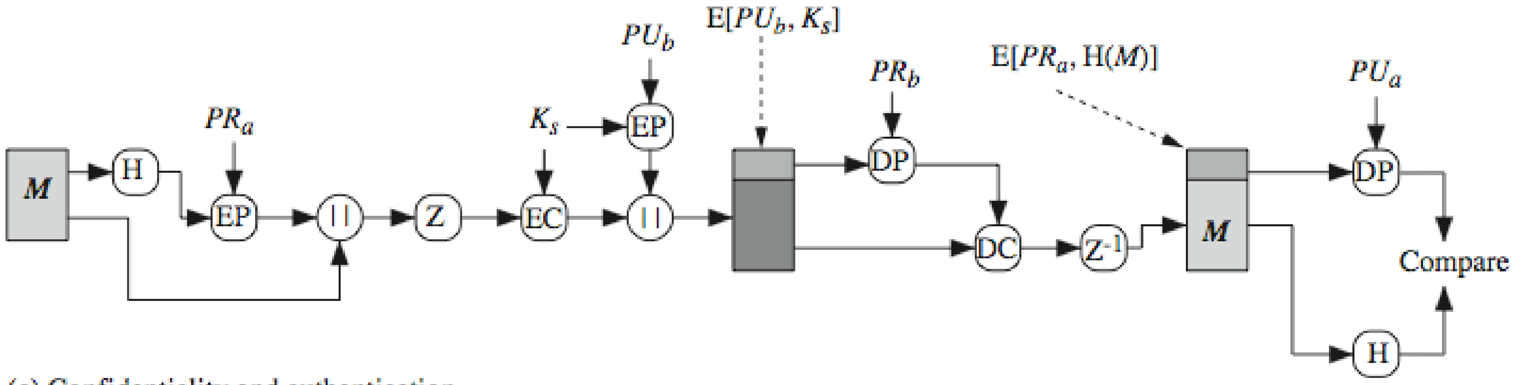
1. Sender forms 128-bit random session key
2. Encrypts message with session key
3. Attaches session key encrypted with RSA
4. Receiver decrypts & recovers session key
5. Session key is used to decrypt message



(b) Confidentiality only

Confidentiality & Authentication

- Can use both services on same message
  - Create signature & attach to message
  - Encrypt both message & signature
  - Attach RSA/ElGamal encrypted session key



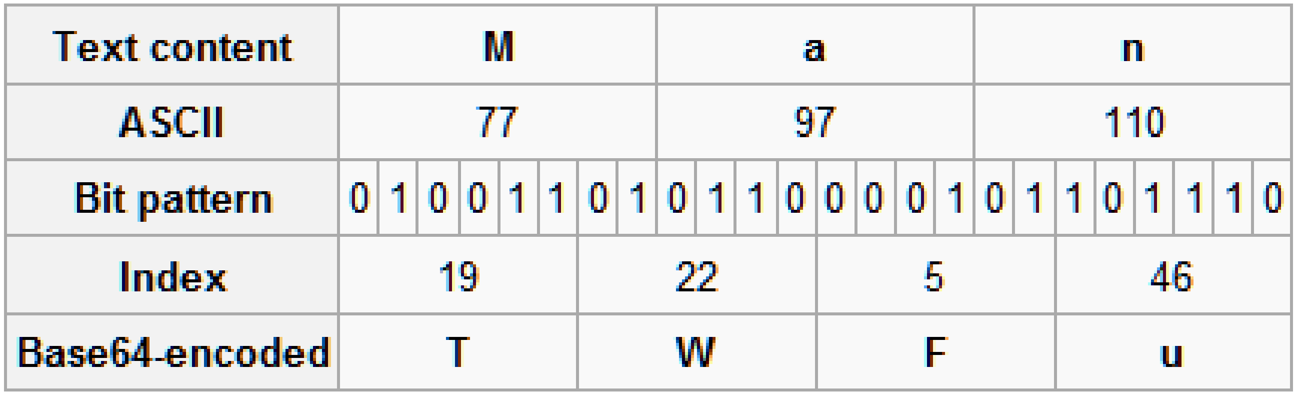
(c) Confidentiality and authentication

PGP Operation – Compression

- By default PGP compresses message after signing but before encrypting
  - Uncompressed message & signature can be stored for later verification
- Compression is non deterministic
  - Uses ZIP compression algorithm

PGP Operation – Email Compatibility

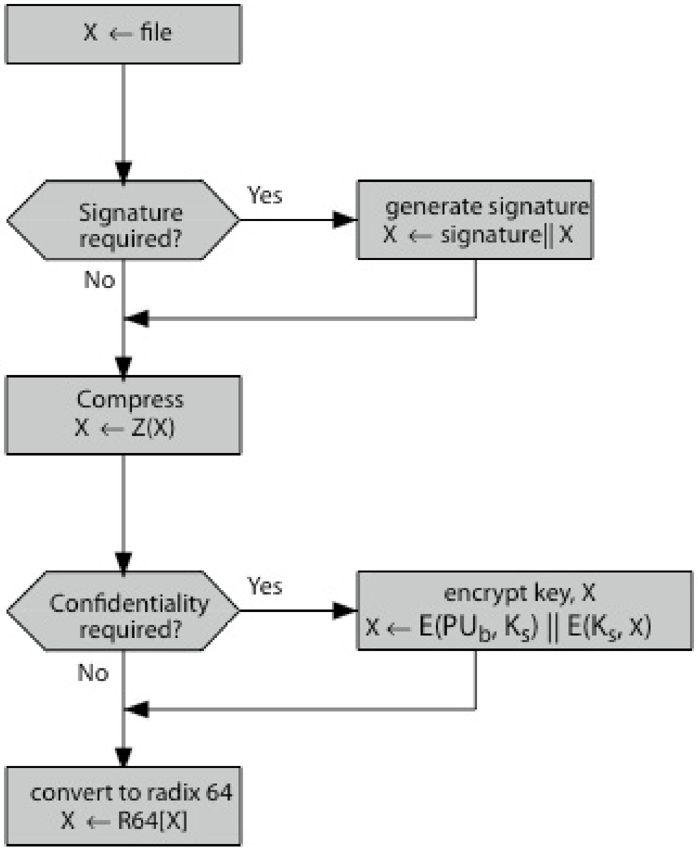
- PGP segments messages if too big
- PGP produces binary (encrypted) data appends a CRC
- Email was designed only for text
  - Need to encode binary into printable ASCII characters
- Uses radix-64 or base-64 algorithm
- Maps 3 bytes to 4 printable chars



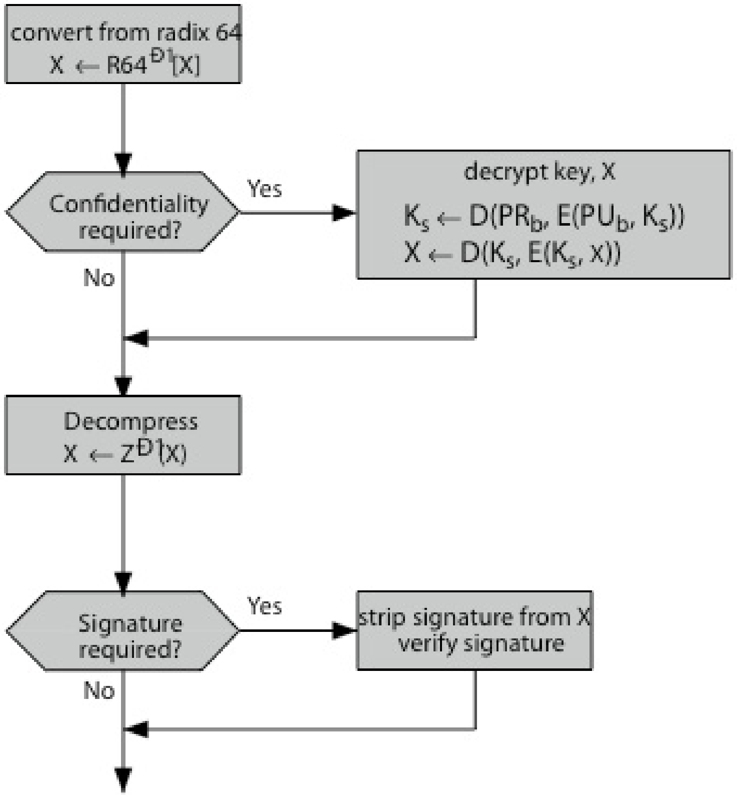
<table>
<tr><th>Text content</th><th colspan="8">M</th><th colspan="8">a</th><th colspan="8">n</th></tr>
<tr><td>ASCII</td><td colspan="8">77</td><td colspan="8">97</td><td colspan="8">110</td></tr>
<tr><td>Bit pattern</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td>Index</td><td colspan="6">19</td><td colspan="6">22</td><td colspan="6">5</td><td colspan="6">46</td></tr>
<tr><td>Base64-encoded</td><td colspan="6">T</td><td colspan="6">W</td><td colspan="6">F</td><td colspan="6">u</td></tr>
</table>

Ref: http://en.wikipedia.org/wiki/Base64

PGP Operation – Summary



(a) Generic Transmission Diagram (from A)



(b) Generic Reception Diagram (to B)

PGP Session Keys

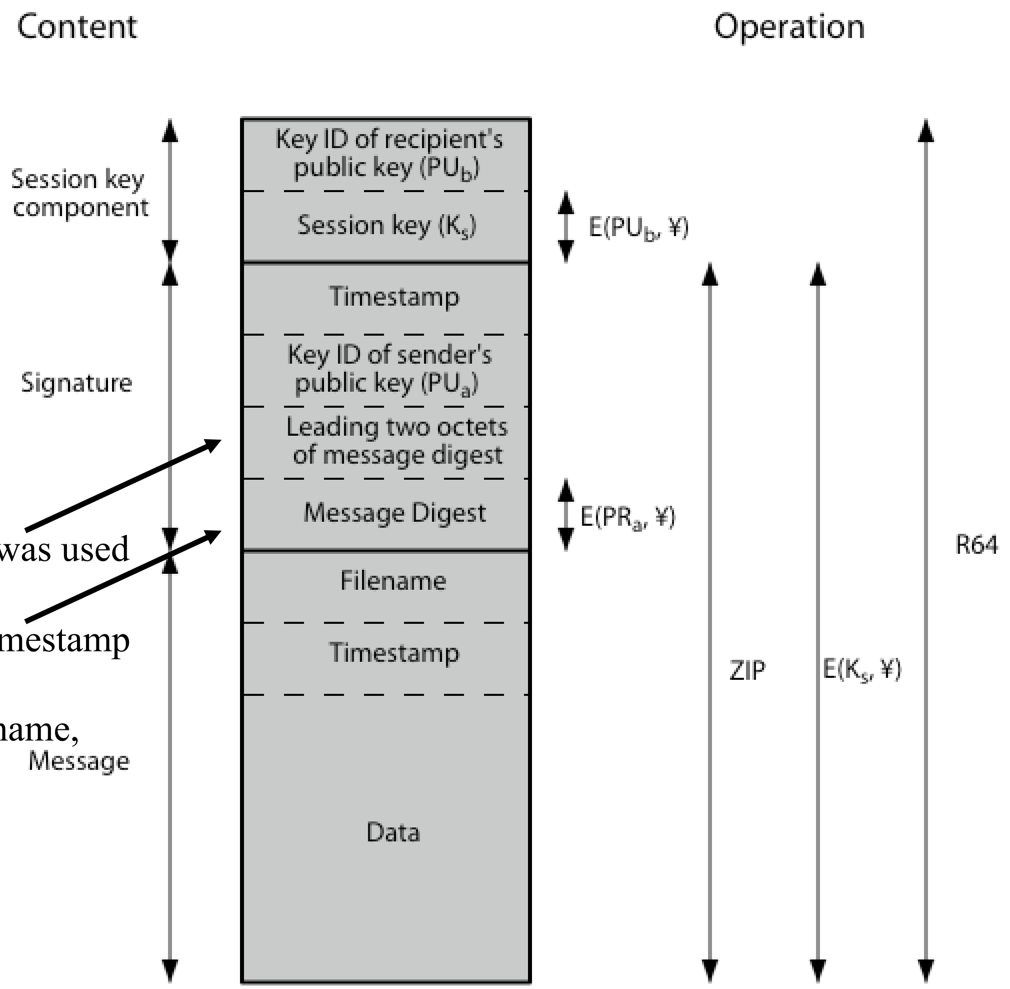
- Need a session key of varying sizes for each message:
  - 56-bit DES,
  - 168-bit Triple-DES
  - 128-bit CAST (Carlisle Adams and Stafford Tavares)
  - IDEA (International Data Encryption Algorithm)
- Generated with CAST-128 using random inputs taken from previous uses and from keystroke timing of user

Ref: http://en.wikipedia.org/wiki/CAST-128 , http://en.wikipedia.org/wiki/Idea_encryption

PGP Public & Private Keys

- Users are allowed to have multiple public/private keys
  ⇒ Need to identify which key has been used
  - Use a key identifier = Least significant 64-bits of the key
- Signature keys are different from encryption keys
  (Encryption keys may need to be disclosed for legal reasons)

PGP Message Format

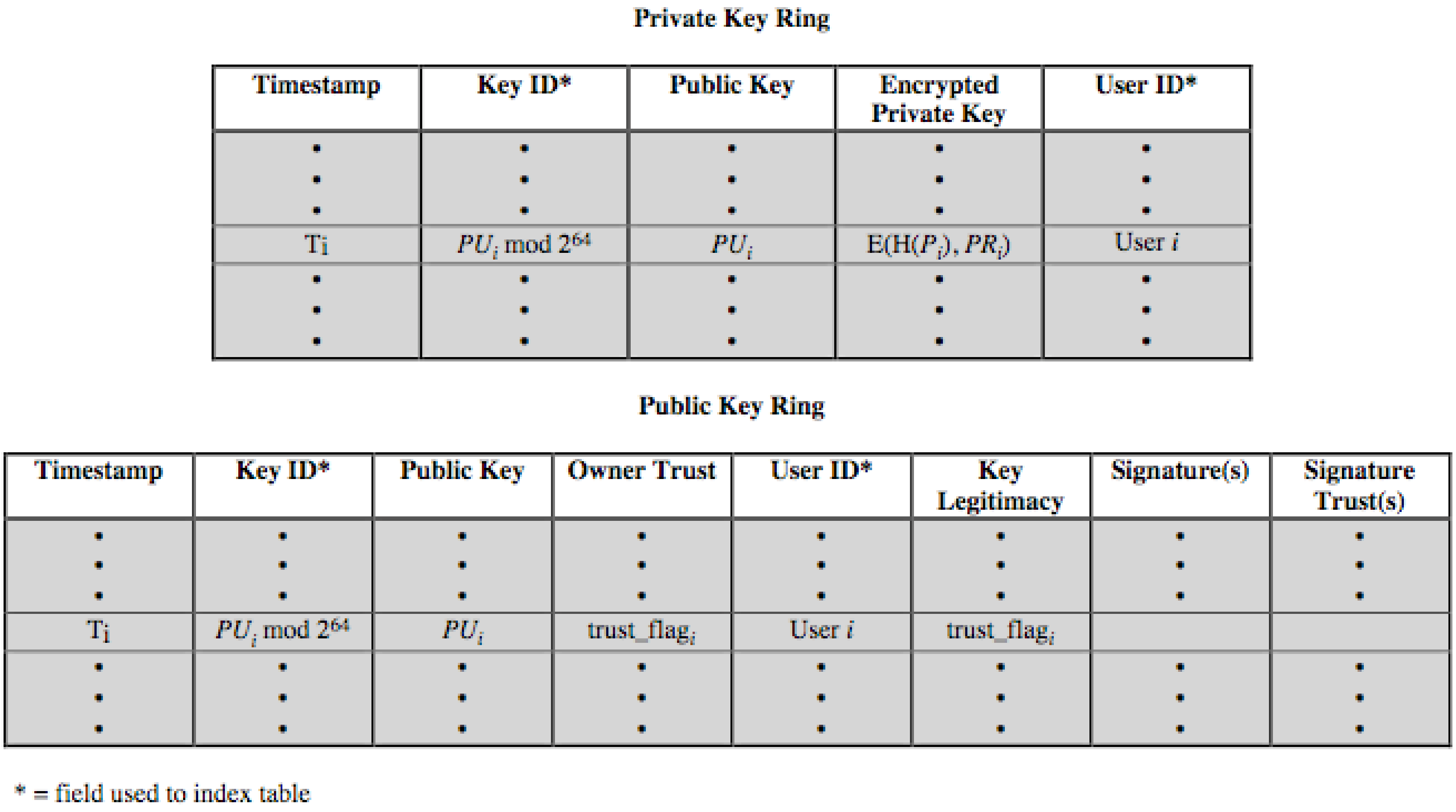


Confirms correct key was used

Includes Signature, timestamp of signature, data
Does not include filename, timestamp of file

PGP Key Rings

- Private keys encrypted by a passphrase
- Public keys of all correspondents



Private Key Ring

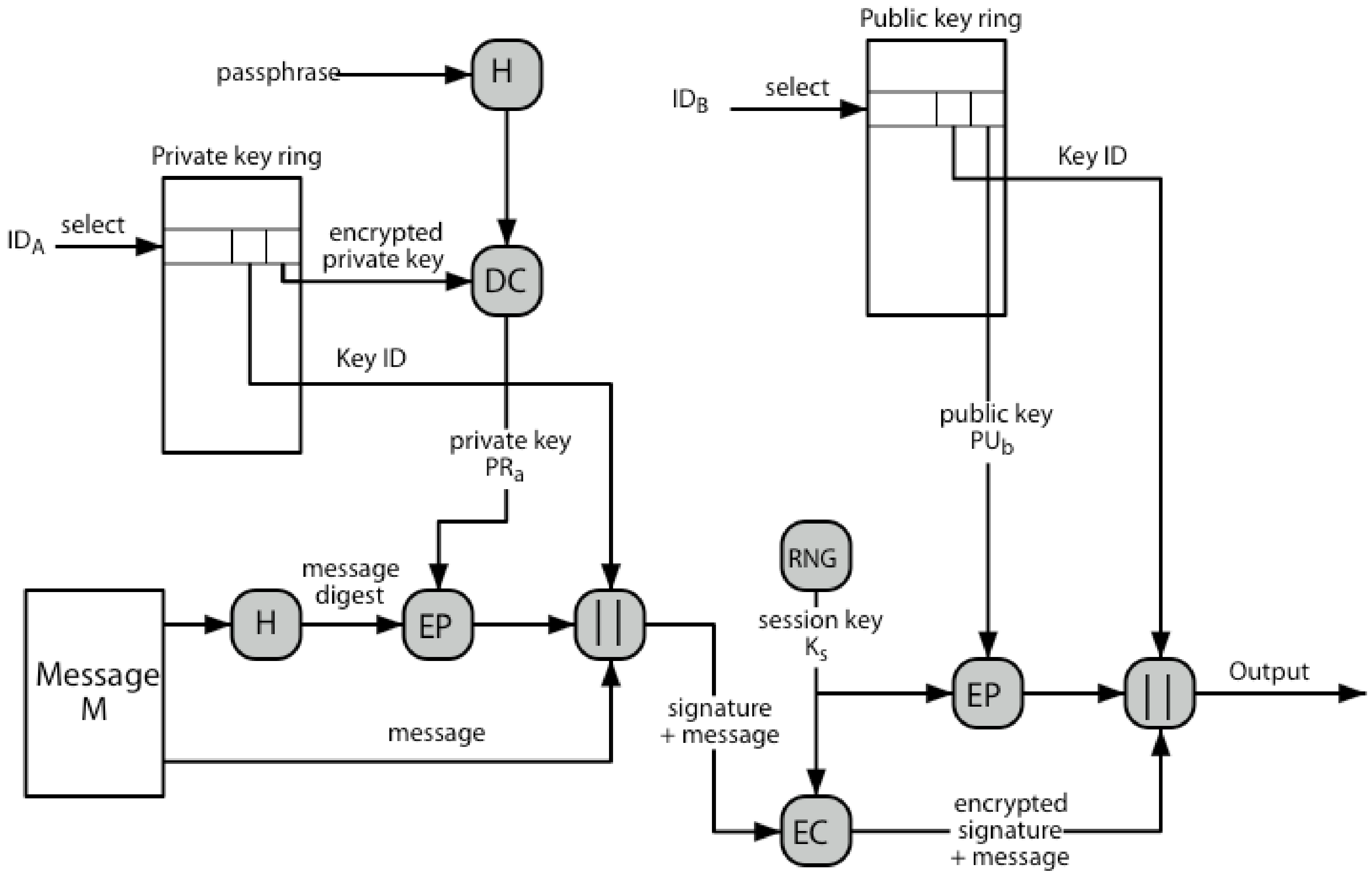
Timestamp	Key ID*	Public Key	Encrypted Private Key	User ID*
• • •	• • •	• • •	• • •	• • •
Ti	PU_i mod 2^{64}	PU_i	E(H(P_i), PR_i)	User i
• • •	• • •	• • •	• • •	• • •

Public Key Ring

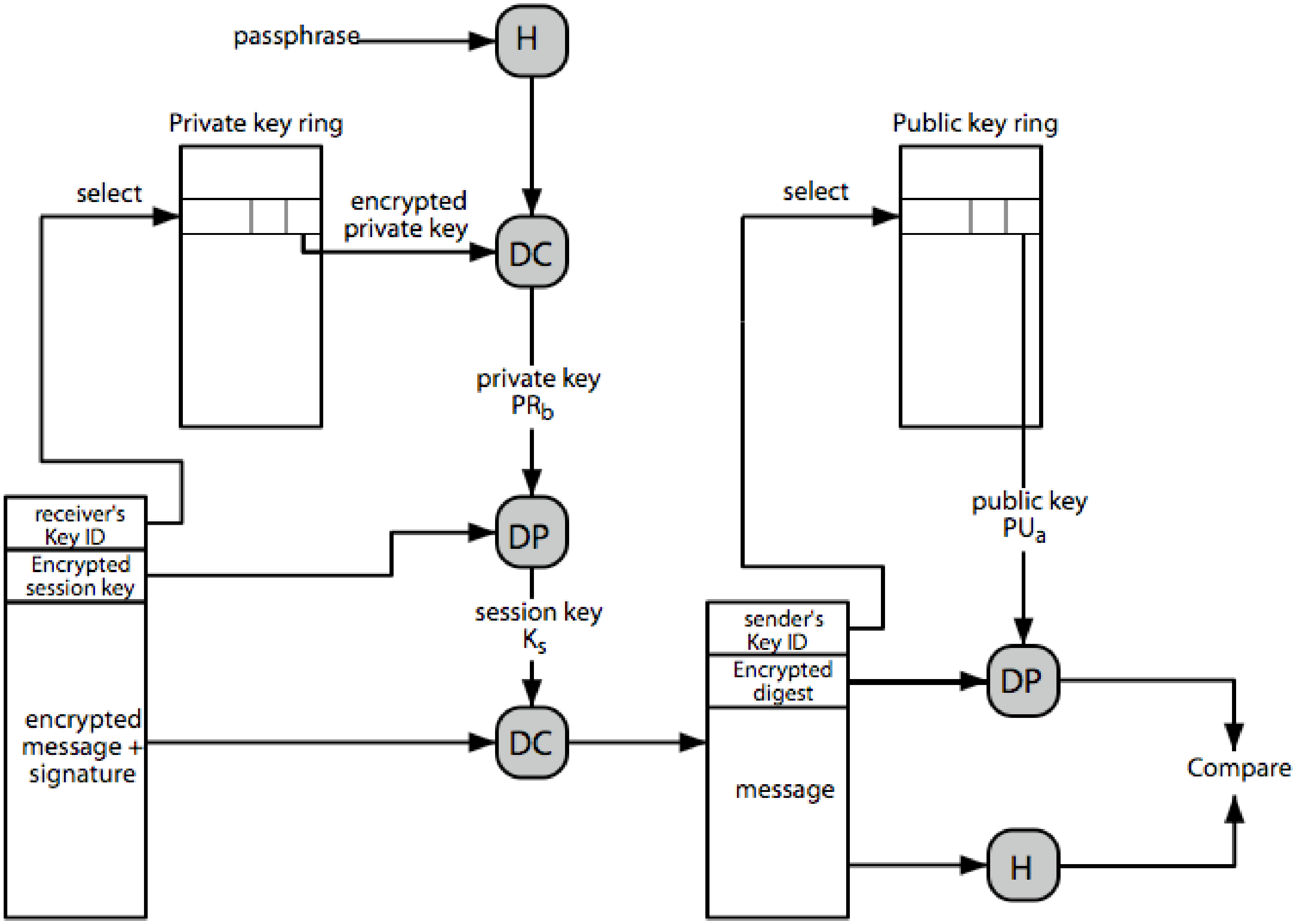
Timestamp	Key ID*	Public Key	Owner Trust	User ID*	Key Legitimacy	Signature(s)	Signature Trust(s)
• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •
Ti	PU_i mod 2^{64}	PU_i	trust_flag$_i$	User i	trust_flag$_i$		
• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •

* = field used to index table

PGP Message Generation



PGP Message Reception

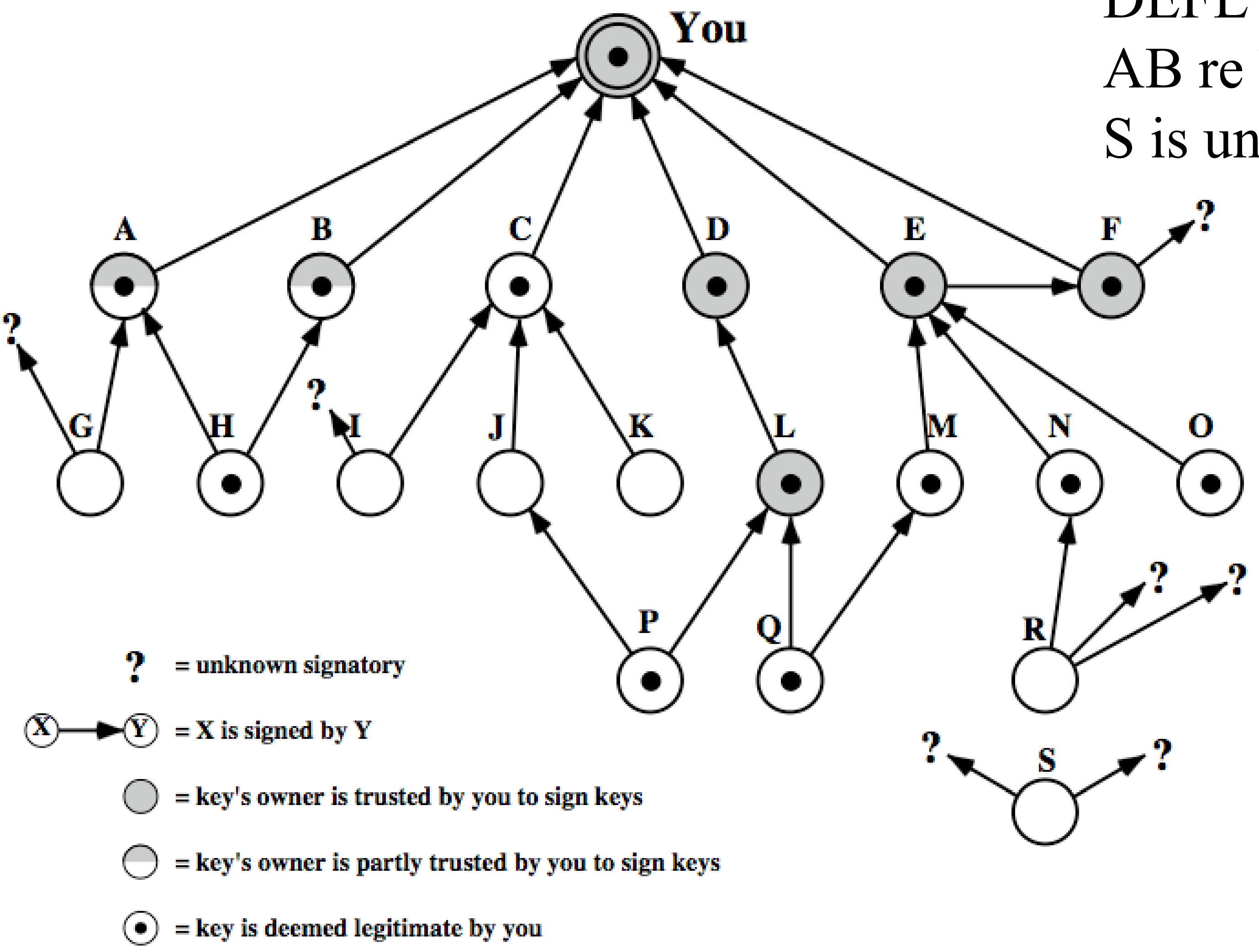


Web of Trust

- There is no need to buy certificates from companies
- A user can sign other user's certificates
- If you trust someone, you can trust users that they sign for.
- You can assign a level of trust to each user and hence to the certificate they sign for
- For example,
  - A certificate that is signed by a fully trusted user is fully trusted
  - A certificate signed by two half trusted users is fully trusted
  - A certificate signed by one half trusted user is half trusted
  - Some certificates are untrusted.

Ref: http://en.wikipedia.org/wiki/Web_of_trust

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

DEFL are trusted
AB re half trusted
S is untrusted

Certificate Revocation

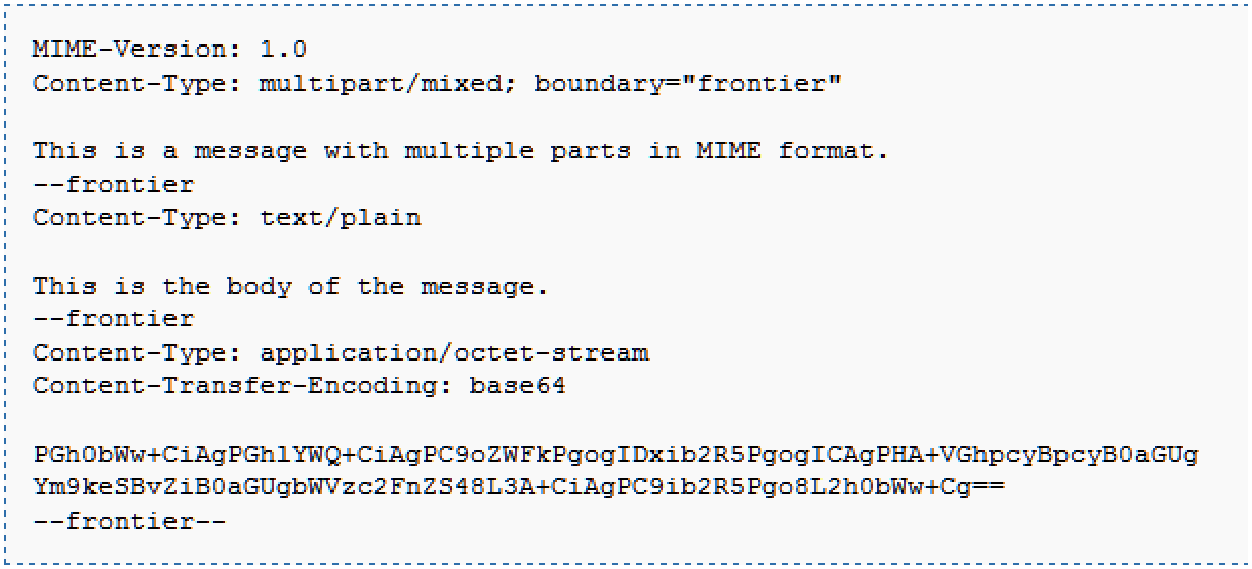
- Owners can revoke public key by issuing a “revocation” certificate signed with the revoked private key
- New Web-of-trust certificates have expiry dates

S/MIME

- Secure/Multipurpose Internet Mail Extensions
- Original Internet RFC822 email was text only
- MIME for varying content types and multi-part messages
  - With encoding of binary data to textual form
- S/MIME added security enhancements
  - Enveloped data: Encrypted content and associated keys
  - Signed data: Encoded message + signed digest
  - Clear-signed data: Clear text message + encoded signed digest
  - Signed & enveloped data: Nesting of signed & encrypted entities
- Have S/MIME support in many mail agents
  - E.g., MS Outlook, Mozilla, Mac Mail etc

MIME Functions

- Types: Text/Plain, Text/Enriched, Multipart/Mixed, Image/jpeg, Image/gif, Video/mpeg, audio/basic, …
- Encodings: 7bit, 8bit, binary, quoted-printable, base64



```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="frontier"

This is a message with multiple parts in MIME format.
--frontier
Content-Type: text/plain

This is the body of the message.
--frontier
Content-Type: application/octet-stream
Content-Transfer-Encoding: base64

PGh0bWw+CiAgPGhlYWQ+CiAgPC9oZWFkPgogIDxib2R5PgogICAgPHA+VGhpcyBpcyB0aGUg
Ym9keSBvZiB0aGUgbWVzc2FnZS48L3A+CiAgPC9ib2R5Pgo8L2h0bWw+Cg==
--frontier--
```

- Quoted-Printable: non-alphanumerics by =2 hex-digits, e.g., “=09” for tab, “=20” for space, “=3D” for =

Ref: http://en.wikipedia.org/wiki/MIME, http://en.wikipedia.org/wiki/Quoted-printable http://en.wikipedia.org/wiki/Base64

S/MIME Cryptographic Algorithms

- Digital signatures: DSS & RSA
- Hash functions: SHA-1 & MD5
- Session key encryption: ElGamal & RSA
- Message encryption: AES, Triple-DES, RC2/40 and others
- MAC: HMAC with SHA-1
- Have process to decide which algorithms to use

S/MIME Messages

- S/MIME secures a MIME entity with a signature, encryption, or both
- Forming a MIME wrapped PKCS object
(Public Key Cryptography Standard originally by RSA Inc Now by IETF)

Type	Subtype	Smime parameter	Meaning
Multipart	Signed		clear msg w signature
Application	Pkcs7-mime	signedData	Signed entity
Application	Pkcs7-mime	envelopedData	Encrypted entity
Application	Pkcs7-mime	Degenerate signedData	Certificate only
Application	Pkcs7-mime	CompressedData	Compressed entity
Application	Pkcs7-signature	signedData	Signature

```
Content-Type: application/pklcs7-mime; smime-type=signedData; name=smime.p7m
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename=smime.p7m
```

Ref: http://en.wikipedia.org/wiki/PKCS

S/MIME Certificate Processing

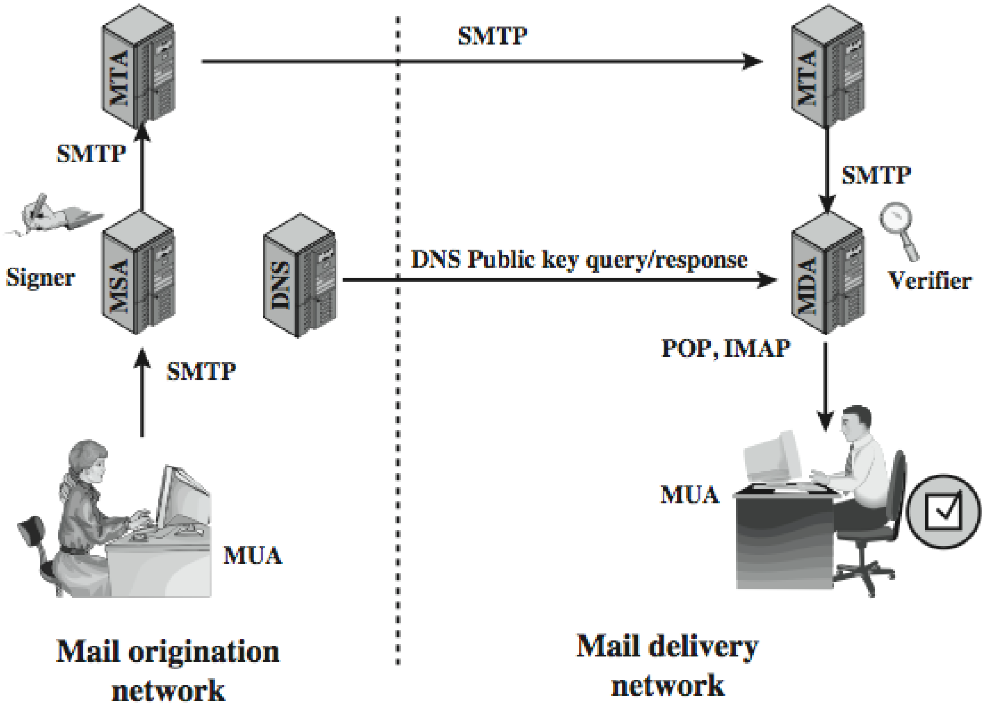
- S/MIME uses X.509 v3 certificates
- Managed using a hybrid of a strict X.509 CA hierarchy and enterprise's CAs
- Each client has a list of trusted CA's certificates and his own public/private key pairs & certificates
- Several types of certificates with different levels of checks:
- Class 1: Email and web browsing
- Class 2: Inter-company email
- Class 3: Banking, …

S/MIME Enhanced Security Services

- RFC2634 (1999) describes enhanced security services:
  - Signed receipts: Request a signed receipt
  - Security labels: Priority, which users (role) can access
  - Secure mailing lists: Request a list processor to encrypt

Domain Keys Identified Mail

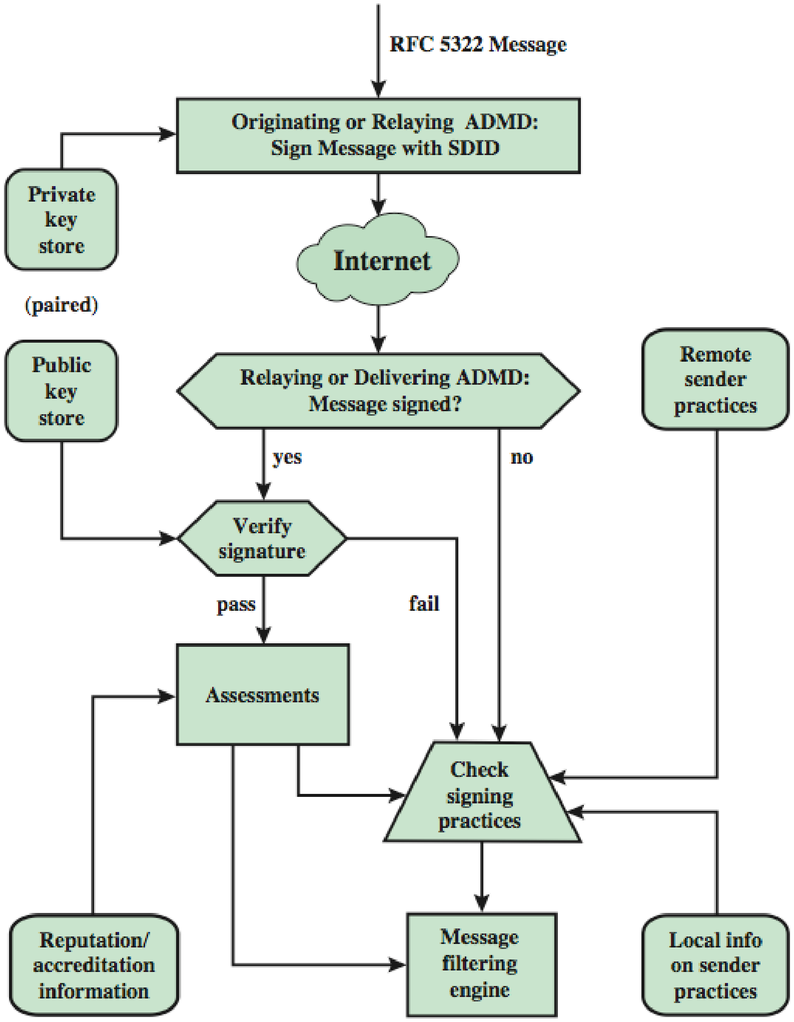
- Emails signed by the enterprise, e.g. WUSTL rather than the sender
- Company's mail system signs the message
- So spammers cannot fake that companies email addresses



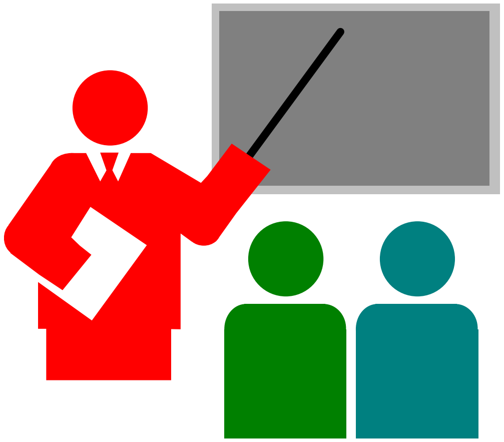
DNS = domain name system
MDA = mail delivery agent
MSA = mail submission agent
MTA = message transfer agent
MUA = message user agent

Ref: http://en.wikipedia.org/wiki/DKIM

DCIM Functional Flow



Summary



1. Email can be signed, encrypted or both
2. PGP is a commonly used system that provides integrity, authentication, privacy, compression, segmentation, and MIME compatibility
3. PGP allows Web of trust in addition to CA certificates
4. S/MIME extends MIME for secure email and provides authentication and privacy
5. DKIM allows originating companies to sign all emails from their users

Homework 18

- A. [18.4] The first 16 bits of the message digest in a PGP signature are transmitted in the clear. To what extent does this compromise the security of the hash algorithm?
- B. [18.9] Encode the text “plaintext” using Radix-64 and quoted-printable