Chapter 5

Electronic mail security

Source by Henric Johnson

Blekinge Institute of Technology, Sweden

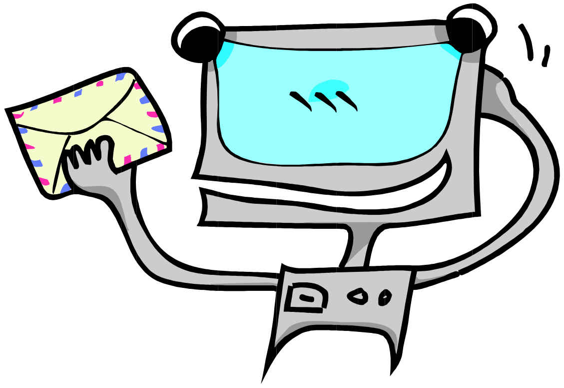
http://www.its.bth.se/staff/hjo/

Henric.Johnson@bth.se

Outline

- Pretty good privacy
- S/MIME
- Recommended web sites

Pretty Good Privacy



Pretty Good Privacy

- The creator of PGP: Philip R. Zimmerman
- PGP provides a **confidentiality** and **authentication service**
  - Applications: Electronic mail and file storage
- What has done by Zimmermann?
  - Page 122-123

Why Is PGP Popular?

- It is availiable free on a variety of platforms.
- Based on well known algorithms.
  - Public-key encryption: RSA, DSS, Diffie-Hellman
  - Symmetric encryption: CAST-128, IDEA, 3DES
  - Hash coding: SHA-1
- Wide range of applicability
- Not developed or controlled by governmental or standards organizations.
- PGP is now on an Internet standards track (RFC 3156).

Operational Description

- Consist of five services:
  - Authentication
  - Confidentiality
  - Compression
  - E-mail compatibility
  - Segmentation
- Notation
  - Page 123

Authentication

- Digital signature
  - Algorithms used: DSS/SHA or RSA/SHA
- **Detached signatures** are supported.
  - A detached signature may be stored and transmitted separately from the message it signs.
  - Applications
    - A user may wish to maintain a separate signature log of all messages sent or received.
    - A detached signature of an executable program can detect subsequent virus infection.
    - Used for more than one party must sign a document.

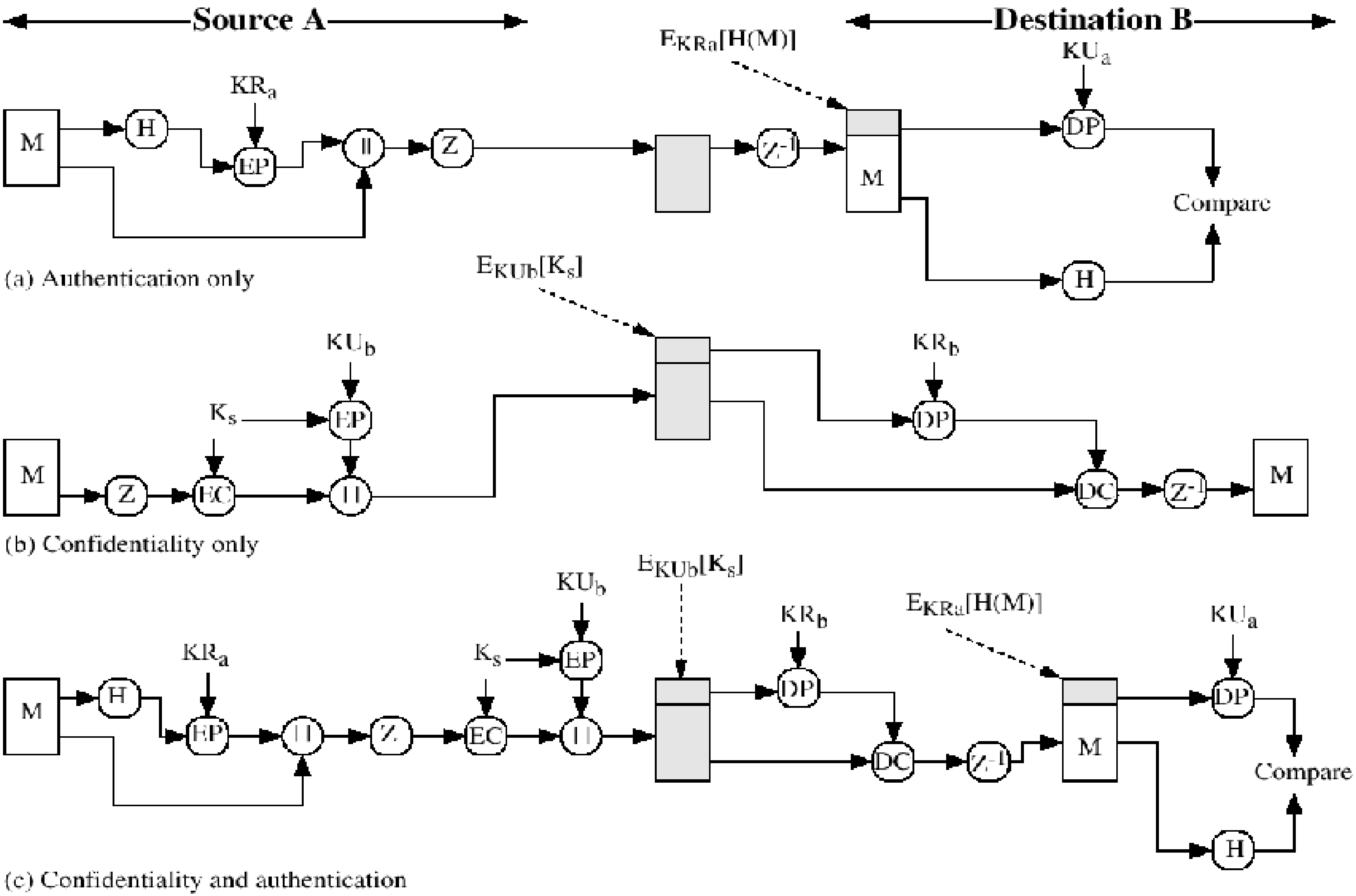


Figure 5.1 PGP Cryptographic Functions

Confidentiality

- Message encryption
  - Algorithms used: CAST or IDEA or 3-key 3DES with Diffie-Hellman or RSA
    - ElGamal: A variant of Diffie-Hellman that does provide encryption/decryption.
- Key distribution: **One-time key**
  - Each symmetric key is used only once.
  - A new key is generated as a random 128-bit number for each message.
  - It is encrypted with the receiver's public key.

Confidentiality

- Observations
  - To reduce encryption time the combination of symmetric and public-key encryption is used.
  - The use of the public-key algorithm solves the session key distribution problem.
    - Each message is a one-time independent event with its own key.
    - Given the store-and-forward nature of electronic mail, the use of handshaking to assure that both sides have the same session key is not practical.
  - The use of one-time symmetric keys strengthens what is already a strong symmetric encryption approach.
  - PGP provides the user with a range of key size options from 768-3072 bits.
    - DSS key for signatures is limited to 1024 bits.

Confidentiality and Authentication

- Figure 5.1c
- **This sequence is preferable to the opposite: encrypting the message and then generating a signature for the encrypted message.**
- It is generally more convenient to store a signature with a plaintext version of a message.
- For purposes of third-party verification, if the signature is performed first, a third party need not be concerned with the symmetric key when verifying the signature.

Compression

- **PGP compresses the message after applying the signature but before encryption.**
  - The benefit of saving space both for e-mail transmission and for file storage.
- The placement of the compression algorithm is critical.
- The compression algorithm used is ZIP.
- Two reasons about the signature is generated before compression.
  - Page 127

E-mail Compatibility

- Many electronic mail systems only permit the use of blocks consisting of ASCII text.
- When PGP is used
  - Part or all of the resulting block consists of a stream of arbitrary 8-bit octets.
  - Raw 8-bit binary stream → A stream of printable ASCII characters
    - **Radix-64 conversion**
  - The use of radix-64 expands the message by 33%.

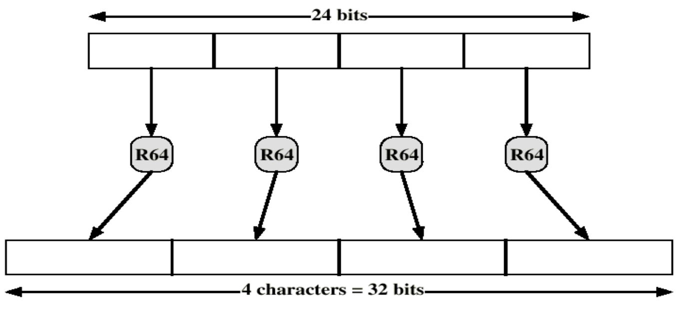
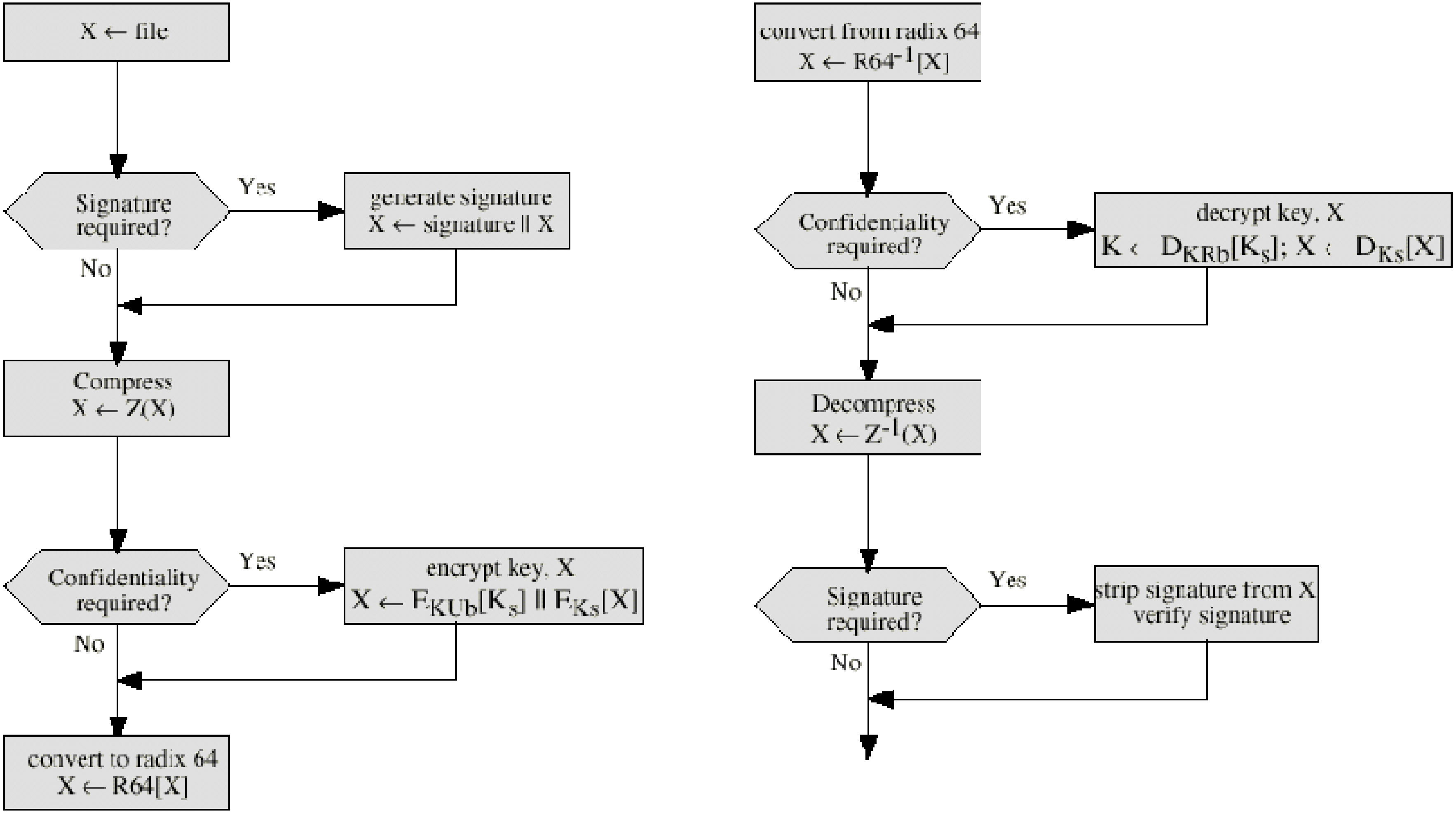


Figure 5.11 Printable Encoding of Binary Data into Radix-64 Format

E-mail Compatibility

- The radix-64 algorithm
  - It blindly converts the input stream to radix-64 format regardless of content, even if the input happens to be ASCII text.
    - Providing a certain level of confidentiality.
  - An option
    - Only used for the signature portion of signed plaintext messages.
    - This enables the human recipient to read the message without using PGP.



(a) Generic Transmission Diagram (from A)

(b) Generic Reception Diagram (to B)

Figure 5.2 Transmission and Reception of PGP Messages

Segmentation and Reassembly

- 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 segmentation is done after all of the other processing, including the radix-64 conversion.
- The receiver strip of all e-mail headers and reassemble the block.
  - The session key component and signature component appear only once, at the beginning of the first segment.

Cryptographic Keys and Key Rings

- Use of four types of keys
  - One-time session symmetric keys
  - Public keys
  - Private keys
  - **Passphrase-based symmetric keys**

- Three separate requirements can be identified with respect to these key.
  - Page 130

Session Key Generation

- Each session key is associated with a single message and is used only for the purpose of encryption and decryption that message.
- Example: CAST-128 (Page 130)
  - The input to the random number generator
    - A 128-bit key (previous session key)
    - Two 64-bit blocks
      - Based on keystroke input from the user
        - Keystroke timing, the actual keys struck
  - The result is to produce a sequence of session keys that is effectively unpredictable.

Key Identifiers

- Any given user may have multiple public/private key pairs.
- Problem
  - How does the recipient know which of its public keys was used to encrypt the session key?
  - How does the recipient know which of sender’s private keys was used to signed the message?
- Simple solution
  - It is to transmit the public key with the message.
  - Drawback
    - It is unnecessarily wasteful of space.

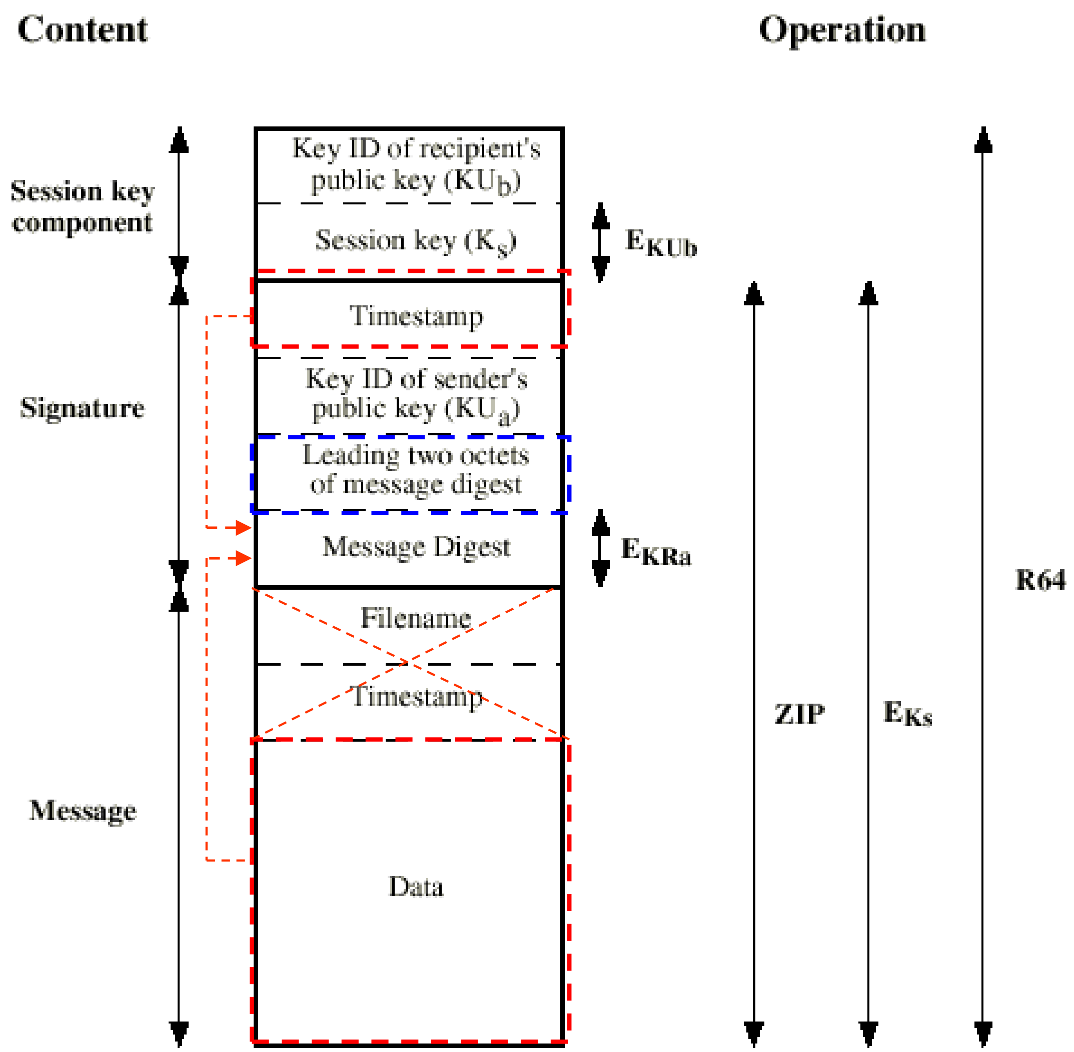
Key Identifiers

- Another solution
  - It would be to associate an identifier with each public key that is unique at least within one user.
  - The combination of user ID and key ID would be sufficient to identify a key uniquely.
  - Only the much shorter key ID would need to be transmitted.
  - Drawback: It raises a management and overhead problem.
    - Key IDs must be assigned and stored so that both sender and recipient could map from key ID to public key. $\Rightarrow$ Unnecessarily burdensome

Key Identifiers

- The solution adopted by PGP
  - It is to assign a key ID to each public key that is, with very high probability, unique within a user ID.
  - The key ID associated with each public key consists of its least significant 64 bits.
    - The key ID of public key KU_a is ($KU_a \bmod 2^{64}$)
  - A key ID is also required for the PGP digital signature.
    - The digital signature component of a message includes the 64-bit key ID of the required public key.
- A message consists of three components
  - The message component, a signature (optional), a session key component (optional)

Format of PGP Message



Key Rings

- Two key IDs are included in any PGP message that provides both confidentiality and authentication.
- These keys need to be stored and organized in a systematic way for efficient and effective use by all parties.
- The scheme used in PGP is to provide a pair of data structures at each node.
  - **Private-key ring**
    - One to store the public/private key pairs owned by that node.
  - **Public-key ring**
    - One to store the public keys of other users known at this node.

Private-key ring

- The general structure of a private-key ring
  - Timestamp, Key ID, Public key, **Private key**, User ID
  - The private key itself is not stored in the key ring. This key is encrypted using CAST-128 (or IDEA or 3DES).
- **The procedure of the encrypted private key**
  - Page 133
- Retrieve a private key from the private-key ring
  - The user must supply the passphrase.
  - PGP will retrieve the encrypted private key, generate the hash code of the passphrase, and decrypt the encrypted private key using CAST-128 with the hash code.

Private Key Ring

Timestamp	Key ID*	Public Key	Encrypted Private Key	User ID*
•	•	•	•	•
•	•	•	•	•
•	•	•	•	•
T_i	$KU_i \bmod 2^{64}$	KU_i	$E_{H(P_i)}[KR_i]$	User i
•	•	•	•	•
•	•	•	•	•
•	•	•	•	•

Public Key Ring

Timestamp	Key ID*	Public Key	Owner Trust	User ID*	Key Legitimacy	Signature(s)	Signature Trust(s)
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
T_i	$KU_i \bmod 2^{64}$	KU_i	trust_flag$_i$	User i	trust_flag$_i$		
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•

* = field used to index table

Figure 5.4 General Structure of Private and Public Key Rings

Public-key Ring

- The general structure of a public-key ring
  - Timestamp, Key ID, Public key, Owner Trust, User ID,
  - **Key legitimacy, Signature(s), Signature Trust(s)**
- Example: message transmission
  - The steps are performed by the sending PGP entity
    - Page 135
    - Figure 5.5
  - The steps are performed by the receiving PGP entity
    - Page 136
    - Figure 5.6

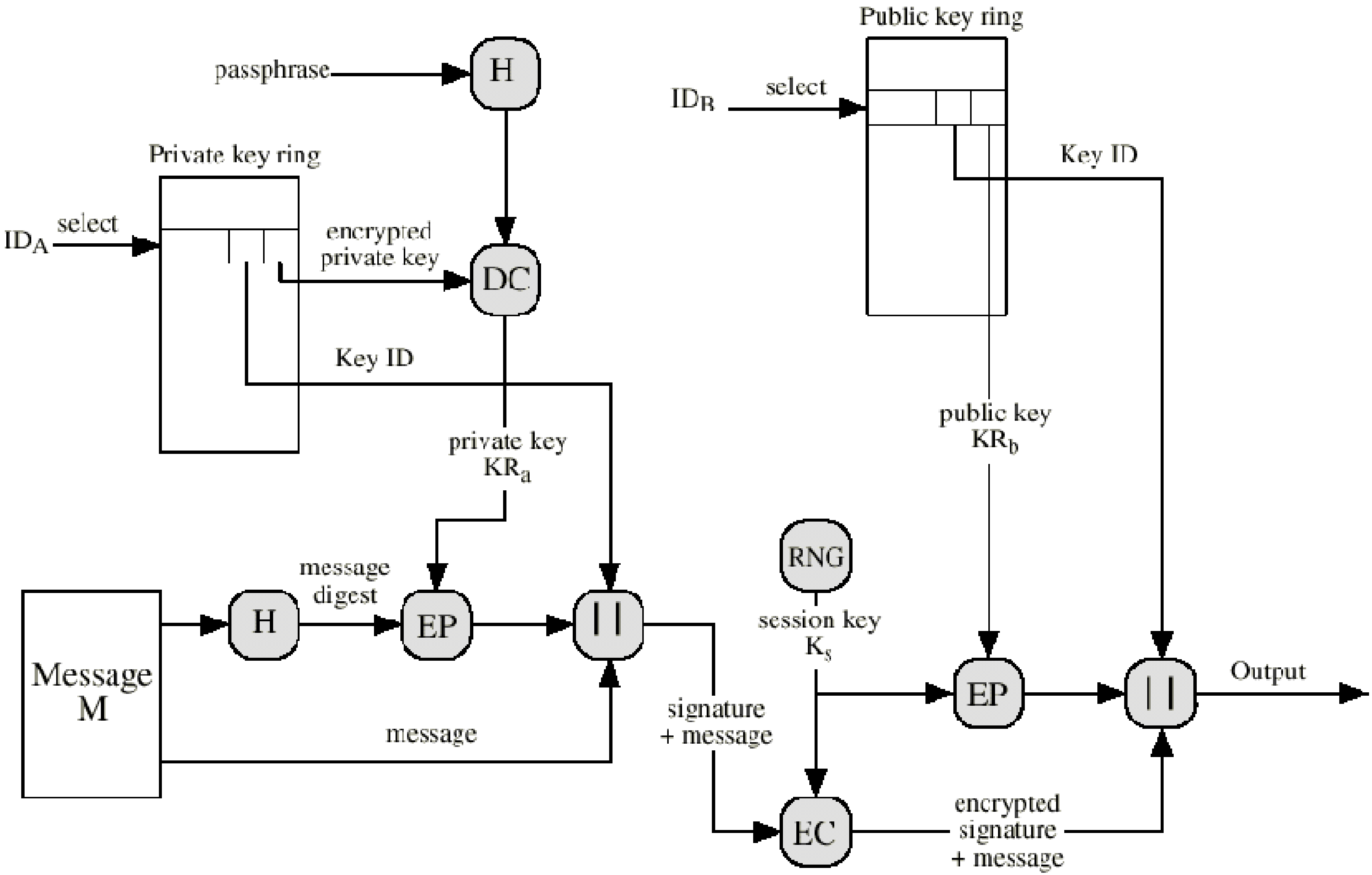


Figure 5.5 PGP Message Generation (from User A to User B; no compression or radix 64 conversion)

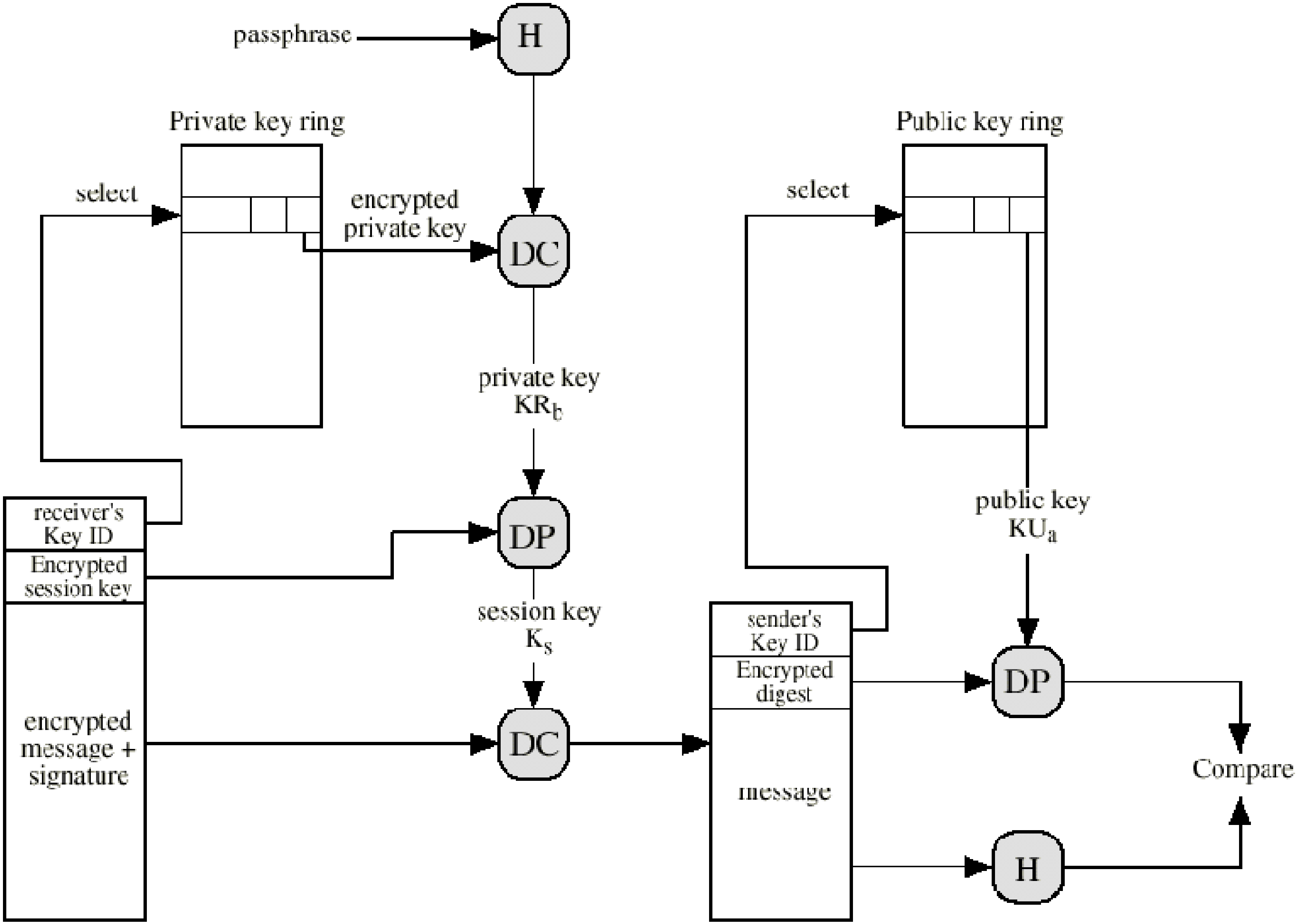


Figure 5.6 PGP Message Reception (from User A to User B; no compression or radix 64 conversion)

Public-Key Management

- Because PGP is intended for use in a variety of formal and informal environments, no rigid public-key management scheme is set up.
- **Approaches to public-key management**
  - The essence of the problem
    - User A must build up a public-key ring containing the public keys of other users to interoperate with them using PGP.
    - Attacking scenario and its two threats are depicted in Page 137.
  - Some approaches minimize the risk that a user's public-key ring contains false public keys.
    - Page 137

The Use of Trust

- No include any specification for establishing certifying authorities or for establishing trust.
  - Solution: The use of trust
    - Associating trust with public keys, and exploiting trust information.
    - **Each entry in the public-key ring is a public-key certificate.**

- The Use of Trust
  - Key legitimacy field
  - Signature trust field
  - Owner trust field

The Use of Trust

- **Key legitimacy field**
  - It indicates the extent to which PGP will trust that this is a valid public key for this user.
    - ***The higher the level of trust, the stronger is the binding of this user ID to this key.***
  - <u>It is computed by PGP.</u>
    - Associated with the entry are zero or more signatures that the key ring owner has collected that sign this certificate.
    - Derived from the collection of signature trust fields in the entry.

The Use of Trust

- **Signature trust field**
  - Each signature has associated with it.
  - It indicates the degree to which this PGP user trusts the signer to certify public keys.
- **Owner trust field**
  - It indicates the degree to which this public key is trusted to sign other public-key certificates.
  - This level of trust is assigned by the user.
- Operations for the use of trust
  - Page 138

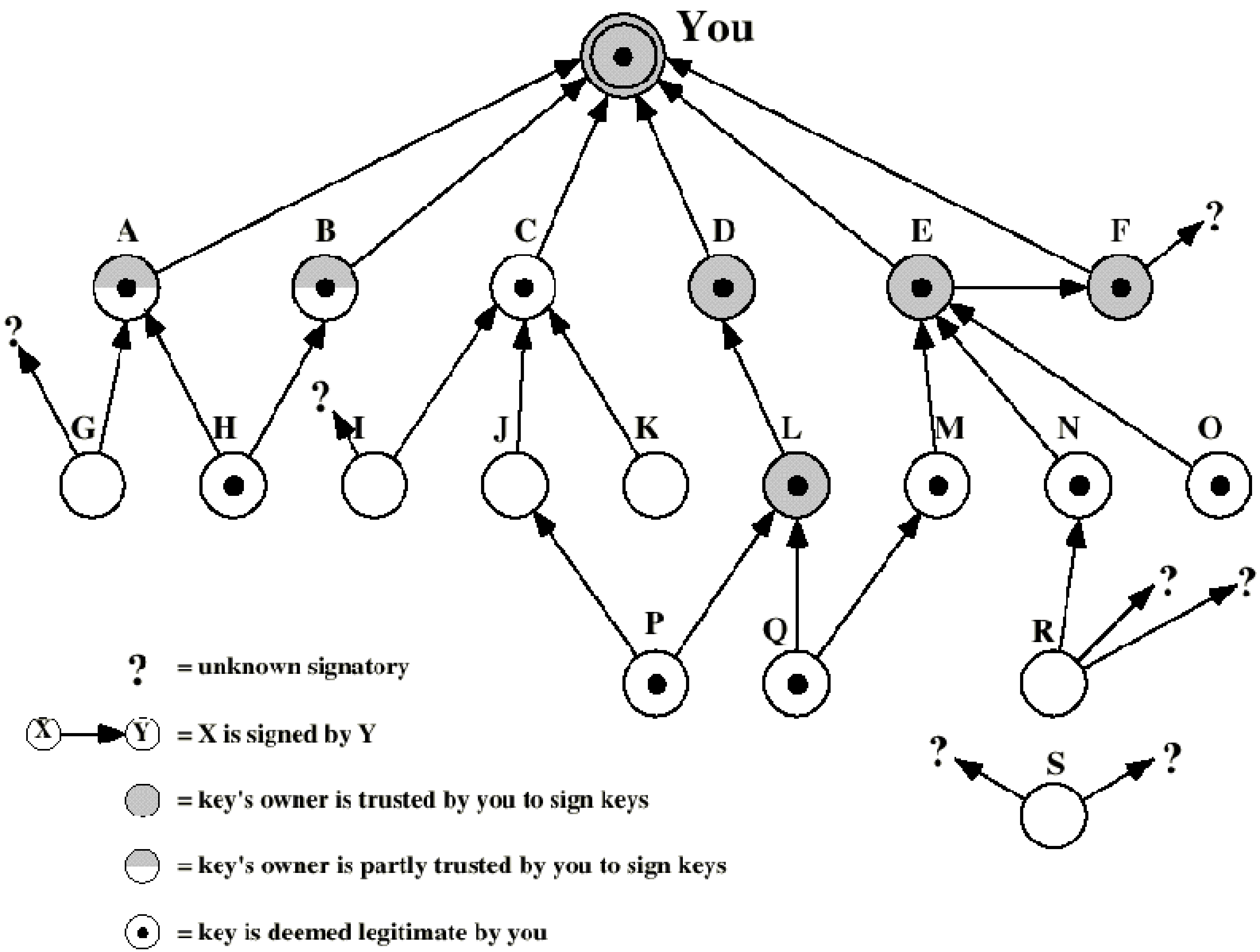
The Use of Trust

- Periodically, PGP processes the public-key ring to achieve consistency.
  - A top-down process
  - For each OWNERTRUST field, PGP scans the ring for all signatures authored by that owner and updates the SIGTRUST field to equal the OWNERTRUST field.
  - This process starts with keys for which there is ultimate trust.
  - Then all KEYLEGIT fields are computed on the basis of the attached signatures.

The Structure of a Public-Key Ring

- Figure 5.7
  - It provides an example of the way in which signature trust and key legitimacy are related.
  - It shows the structure of a public-key ring.
  - The user has acquired a number of public keys, some directly from their owners and some from a third party such as a key server.
- Several points are illustrated in Figure 5.7
  - Page 140-141

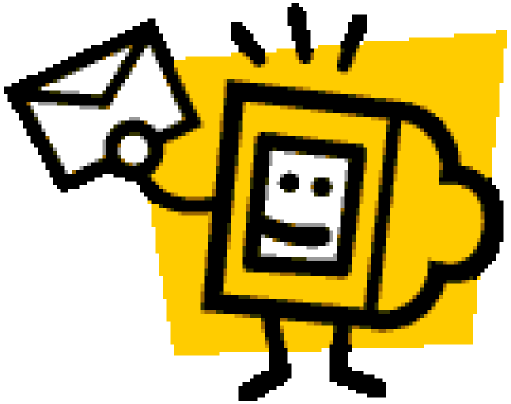
The Structure of a Public-Key Ring



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.

S/MIME (Secure/Multipurpose Internet Mail Extension)



S/MIME

- Secure/Multipurpose Internet Mail Extension
- S/MIME V.S. PGP
  - S/MIME will probably emerge as the industry standard for commercial and organizational use.
  - PGP will remain the choice for personal e-mail security for many users.
- Two prior works for studying S/MIME
  - The underlying e-mail format—MIME
  - The traditional e-mail format standard—RFC 822

Simple Mail Transfer Protocol (SMTP, RFC 822)

- SMTP defines a format for text messages that are sent using electronic mail.
  - Messages are viewed as having an envelope and contents.
- **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)

MIME

- MIME is intended to resolve these problems in a manner that is compatible with existing RFC 822 implementations.
- The elements of MIME
  - Five new message header fields are defined.
  - A number of content formats are defined.
    - Supporting multimedia electronic mail
  - Transfer encodings are defined.

Header fields in MIME

- **MIME-Version**
  - Must be “1.0” → RFC 2045, RFC 2046
- **Content-Type**
  - More types being added by developers (application/word)
- **Content-Transfer-Encoding**
  - How message has been encoded (radix-64)
- **Content-ID**
  - Unique identifying character string.
- **Content Description**
  - Needed when content is not readable text (e.g.,mpeg)

S/MIME Functions

- **Enveloped Data**
  - Encrypted content and encrypted session keys for recipients.
- **Signed Data**
  - Message Digest encrypted with private key of "signer."
- **Clear-Signed Data**
  - Only the digital signature is encoded using base64.
- **Signed and Enveloped Data**
  - Various orderings for encrypting and signing.

Algorithms Used

- **Message Digesting**
  - SHA-1 and MD5
- **Digital Signatures**
  - DSS
- **Secret-Key Encryption**
  - Triple-DES, RC2/40 (exportable)
- **Public-Private Key Encryption** (for session keys)
  - RSA with key sizes of 512 and 1024 bits, and a variant of Diffie-Hellman (ElGamal).

Sending agent

- Two decisions are made by a sending agent
  - The sending agent must determine if the receiving agent is capable of decrypting using a given encryption algorithm.
  - If the receiving agent is only capable of accepting weakly encryption, the sending agent must decide if it is acceptable to send using weak encryption.
- The sending rules
  - Page 151

User Agent Role

- S/MIME uses Public-Key Certificates - X.509 version 3 signed by Certification Authority
- Functions:
  - **Key Generation** - Diffie-Hellman, DSS, and RSA key-pairs.
  - **Registration** - Public keys must be registered with X.509 CA.
  - **Certificate Storage** - Local (as in browser application) for different services.
  - **Signed and Enveloped Data** - Various orderings for encrypting and signing.

User Agent Role

- **Example: Verisign (www.verisign.com)**
  - Three levels (classes) of security for public-key certificates
    - **Class 1 Digital IDs**
      - Buyer's email address confirmed by emailing vital info.
    - **Class 2 Digital IDs**
      - Postal address is confirmed as well, and data checked against directories.
    - **Class 3 Digital IDs**
      - Buyer must appear in person, or send notarized documents.

Enhanced Security Services

- Three enhanced security services: (Page 156-158)
  - Signed receipts
  - Security labels
  - Secure mailing lists

Recommended Web Sites

- PGP home page: www.pgp.com
- MIT distribution site for PGP
- S/MIME Charter
- S/MIME Central: RSA Inc.'s Web Site