Chapter 5
Electronic Mail Security

-Pretty Good Privacy (PGP)

-S/MIME

1

Need for E-Mail Security

- E-mail is necessary for
  - E-Commerce
  - Daily communication
- E-Mail is also very public, allowing for access at each point from the sender's computer to the recipient's screen.

2

Threats to E-Mail

- Message interception (confidentiality)
- Message interception (blocked delivery)
- Message interception and subsequent replay
- Message content modification
- Message origin modification
- Message content forgery by an outsider
- Message origin forgery by an outsider
- Message content forgery by recipient
- Message origin forgery by recipient
- Denial of message transmission

3

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.

4

PGP Features

- It is based on the best available cryptographic algorithms (3DES….)
  - Considered very strong and secure
- Mainly used for email and file storage applications
- Independent of governmental organizations
- Messages are automatically compressed

5

Operational Description

- PGP Consists of five services:
  - Authentication
  - Confidentiality
  - Compression
  - E-mail compatibility
  - Segmentation and Reassembly

6

PGP: Authentication steps

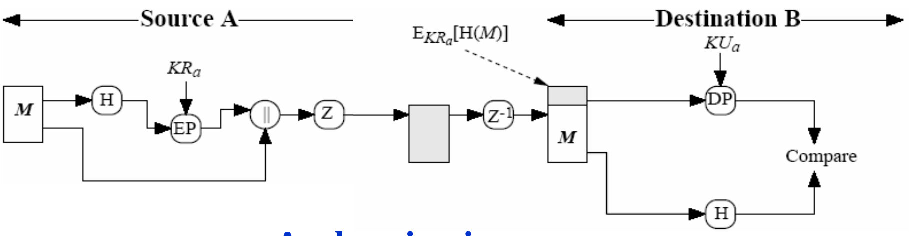
- **Sender:**
  - Creates a message
  - Hashes it to 160-bits using SHA1
  - Encrypts the hash code using her private key, forming a signature
  - Attaches the signature to message

7

PGP: Authentication steps

- **Receiver:**
  - Decrypts attached signature using sender's public key and recovers hash code
  - Recomputes hash code using message and compares with the received hash code'
  - If they match, accepts the message

8



Authentication steps

Stallings, Fig 5.1a

M = original message
H = hash function
| | = concatenation (join)
Z = compression
Z^{-1} = decompression

EP = public key encryption
DP = public key decryption
KR_a = A's private key
KU_a = A's public key

9

PGP: Confidentiality

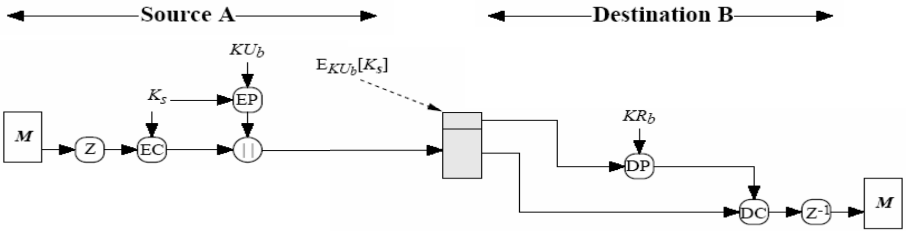
- **Sender:**
  - Generates message and a random number (session key) <u>only</u> for this message
  - Encrypts message with the session key using AES, 3DES, IDEA or CAST-128
  - Encrypts session key itself with recipient's public key using RSA
  - Attaches it to message

10

PGP: Confidentiality

- **Receiver:**
  - Recovers session key by decrypting using his private key
  - Decrypts message using the session key.

11



EC = symmetric encryption
DC = symmetric decryption
K_s = session key
Z = compression
Z^{-1} = Decompression

Confidentiality Stallings, 5.1b

12

Combining authentication and confidentiality in PGP

- Authentication and confidentiality can be combined
  - A message can be both signed **and** encrypted
- This is called **authenticated confidentiality**
- Encryption/Decryption process is "nested" within the process shown for authentication alone

13

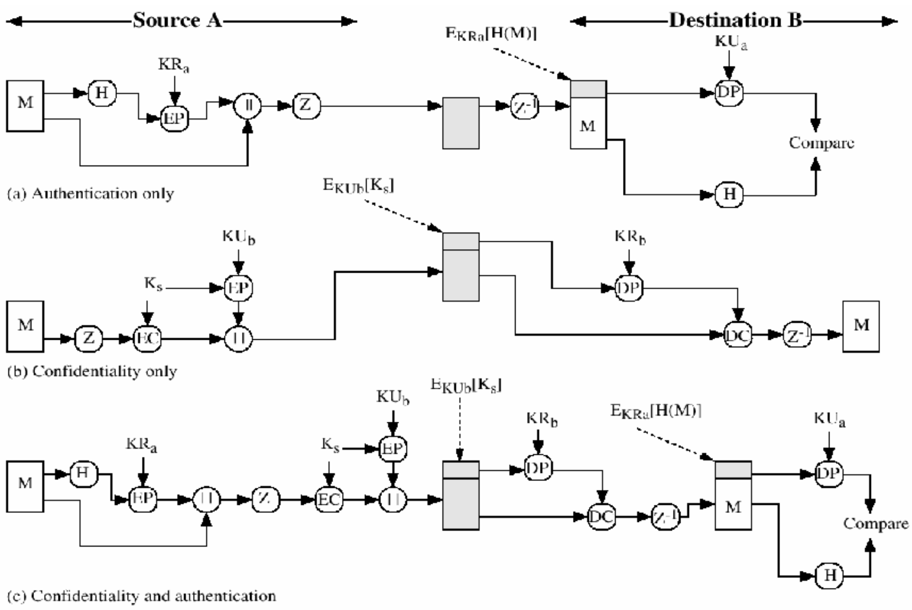


Figure 5.1 PGP Cryptographic Functions

14

Compression

- PGP compresses the message after applying the signature but before encryption
  - Saves space for transmission and storage
- The placement of the compression algorithm is critical.
- The compression algorithm used is ZIP (described in appendix 5A)

15

PGP Compression

- Compression is done after signing the hash. Why?
  - Saves having to compress document every time you wish to verify its signature
- It is also done before encryption. Why?
  - To speed up the process (less data to encrypt)
  - Also improves security
    - Compressed messages are more difficult to cryptanalyze as they have less redundancy

16

PGP Email compatibility

- PGP is designed to be compatible with all email systems
- Handles both the simplest system and the most complex system
- Output of encryption and compression functions is divided into 6-bit blocks
  - Each block is mapped onto an ASCII Character
  - This is called RADIX-64 encoding
  - Has the side-effect of increasing the size of the data by about 33%

17

E-mail Compatibility

The scheme used is radix-64 conversion
(see appendix 5B).
The use of radix-64 expands the message by 33%.

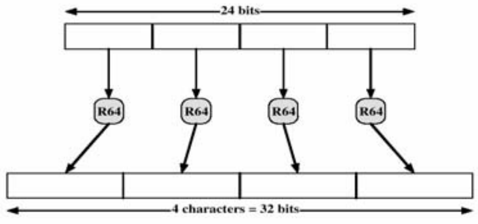


Figure 5.11 Printable Encoding of Binary Data into Radix-64 Format

18

RADIX-64 encoding

6-bit value	character encoding	6-bit value	character encoding	6-bit value	character encoding	6-bit value	character encoding
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/
						(pad)	=

19

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.
- Segementation is done after all other processing
- The receiver strips off all e-mail headers and reassemble the block.

20

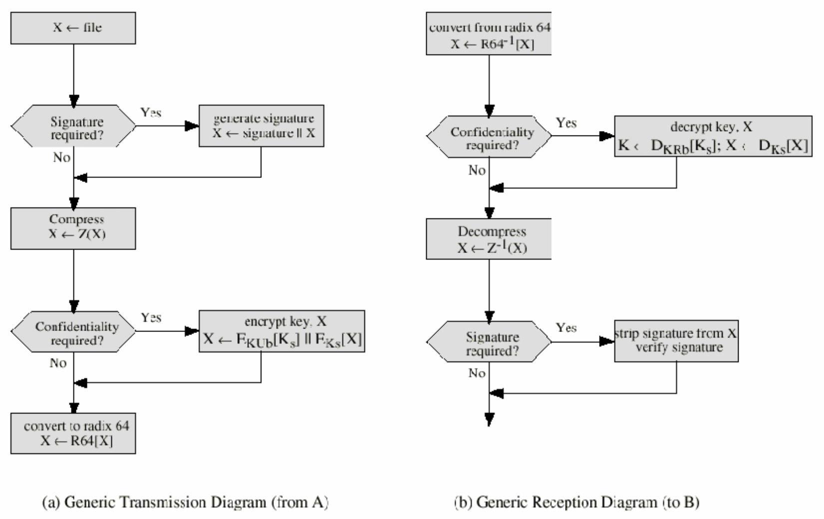


Figure 5.2 Transmission and Reception of PGP Messages

21

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	Split messages into segments

22

Cryptographic Keys and Key Rings

- PGP makes use of 4 types of keys:
  - One-time session symmetric keys
  - Public keys
  - Private Keys
  - Passphrase-based symmetric Keys
    - for storing your private keys encrypted

23

Key Requirements

- A Means of generating unpredictable session keys is needed
- A user is allowed to have multiple public/private key pairs so there must be a way to identify particular keys
- Each PGP entity must maintain a file of its own public/private key pair as well as those of its correspondents

24

Session keys

- Each session key is associated with a single message and is used only once to encrypt and decrypt that message
- Messsage encryption is done with a symmetric encryption algorithm
  - CAST, IDEA use 128 bit keys
  - 3DES uses a 168 bit key
  - Keystrokes and timing are used to generate a “random” stream, which is combined with previous session key toproduce a new unpredictable one.

25

PGP Key Identifiers

- What is a key identifier
- Consider this:
  - A user may have many public/private key pairs
  - He wishes to encrypt or sign a message using one of his keys
  - How does he let the other party know which key he has used?
  - Attaching the whole public key every time is inefficient
- Solution: Generate a **key identifier** (least significant 64-bits of the key)
  - This will most likely be unique and can also be used for signatures

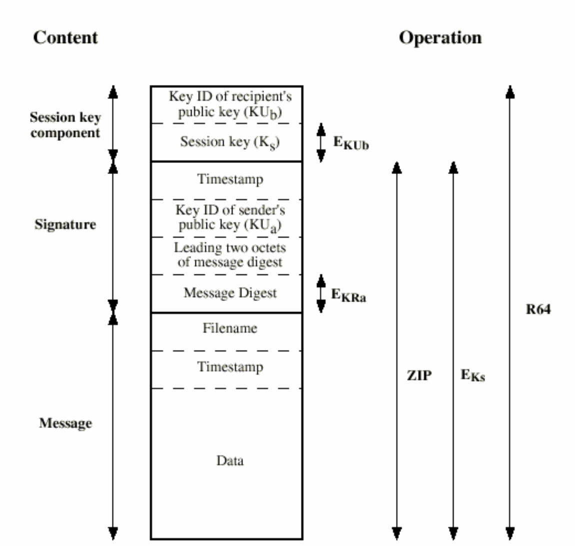
26

Format of PGP Message

- A message may consist of:
- A Message component – data to be stored or transmitted
- A Signature component (optional)
  - Timestamp
  - Message digest encrypted with sender's private signature key
- A Session key (optional)
  - Session key as well as the key used to encrypt the session key
  - ZIPPED and then encoded with radix-64 encoding

27

Format of PGP Message



28

PGP Key Rings

- PGP uses key rings to identify the key pairs that a user **owns** or **trusts**
- Private-key ring contains public/private key pairs of keys he owns
- Public-key ring contains public keys of others he trusts

29

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

30

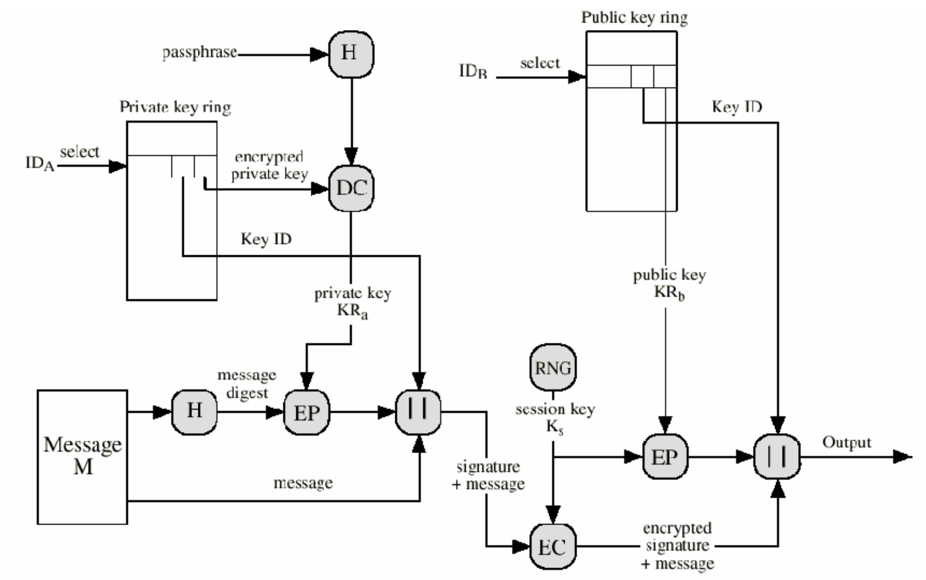


Figure 5.5 PGP Message Generation (from User A to User B; no compression or radix 64 conversion)

31

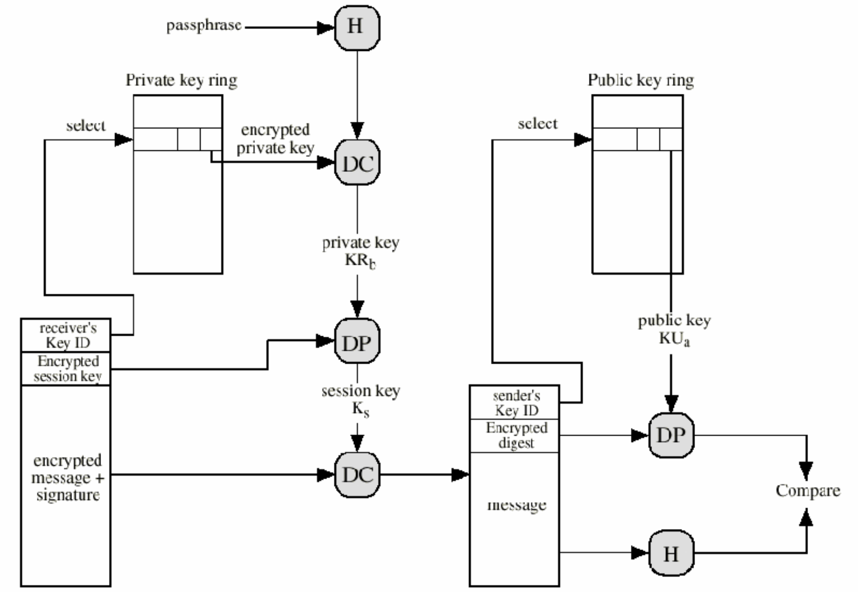


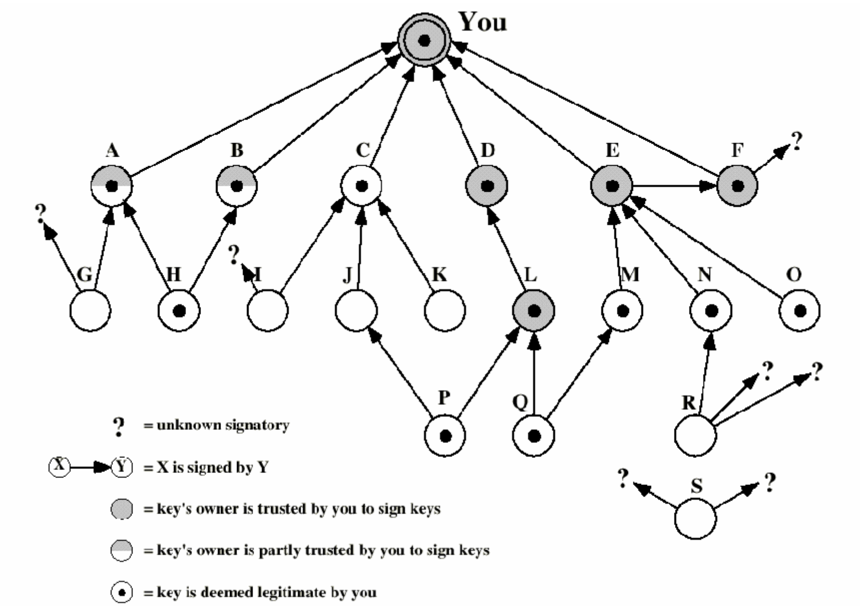
Figure 5.6 PGP Message Reception (from User A to User B; no compression or radix 64 conversion)

32

PGP Public key management

- Key rings are different from certificate chains used in X.509
  - There the user only trusts CAs and the people signed by the CAs
  - Here he or she can trust anyone and can add others signed by people he trusted
- Thus, users do not rely on external CAs
  - A user is his/her own CA

33



34

Revoking Public Keys

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

35

S/MIME

- Secure/Multipurpose Internet Mail Extension
- S/MIME will probably emerge as the industry standard.
- PGP for personal e-mail security

36

RFC 822, 2822

- RFC 822/ 2822:

  RFC 822: **Standard for the format of ARPA Internet text messages**. D. Crocker . Aug-13-1982 (obsoleted by RFC 2822)

  RFC2822: Internet Message Format. P. Resnick, Ed. April 2001.

- In comparison:

  RFC 821: **Simple Mail Transfer Protocol**. J. Postel. Aug-01-1982. (obsoleted by RFC 2821)

  RFC2821: Simple Mail Transfer Protocol. J. Klensin, Ed. April 2001.

37

Limitations of Simple Mail Transfer Protocols (e.g., SMTP, RFC 822)

- **SMTP/822 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: 5 parts (RFCs 2045 through 2049)

38

Header fields in MIME

- **MIME-Version:** Must be "1.0" -> RFC 2045, RFC 2046
- **Content-Type:** More types being added by developers (application/word) See Table 5.3
- **Content-Transfer-Encoding:** How message has been encoded (radix-64) See Table 5.4
- **Content-ID:** (optional) Unique identifying character string.
- **Content Description:** (optional) Needed when content is not readable text (e.g.,mpeg)
- Example MIME message structure: Figure 5.8

39

S/MIME Functions

- **Enveloped Data:** Encrypted content and encrypted session keys for recipients.
- **Signed Data:** Message Digest encrypted with private key of a "signer."
- **Clear-Signed Data:** Signed but not encrypted.
- **Signed and Enveloped Data:** Various orderings for encrypting and signing.

40

Algorithms Used in S/MIME

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

41

New content types in S/MIME

- S/MIME secures a MIME entity with a signature, encryption, or both.
- New types were added for this purpose: SeeTable 5.7
- All of the new application types use the designation PKCS (public key cryptography specifications)

42

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 keypairs.
  - **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.

43

Certificate Security Classes

- **Example: Verisign (www.verisign.com) See Table 5.8**
  - **Class-1:** Buyer's email address confirmed by emailing vital info.
  - **Class-2:** Postal address is confirmed as well, and data checked against directories.
  - **Class-3:** Buyer must appear in person, or send notarized documents.

44

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

45

Additional web sites

- www.pgpi.org
  - the international PGP site (old)
- www.imc.org
  - International mail consortium
- www.openpgp.org
- www.gnupg.org
  - GNU Privacy Guard – Open source PGP

46