UNIT II

E-MAIL SECURITY & FIREWALLS

Electronic Mail Security

- **PGP (**Pretty Good Privacy)
  - Use
    - Digital signature
    - Encryption
    - Compression (zip)
    - Radix-64 conversion

PGP

- Notation

K_s = session key	H = hash function
KP_a = public key of user A	KP_b = public key of user B
KS_a = private key of user A	KS_b = private key of user B
E = conventional encryption	D = conventional decryption
E_p = public-key encryption	D_p = public-key decryption
Z = compression using zip algorithm	Z^{-1} = decompression
$\|\|$ = concatenation	

PGP

Confidentiality via Encryption

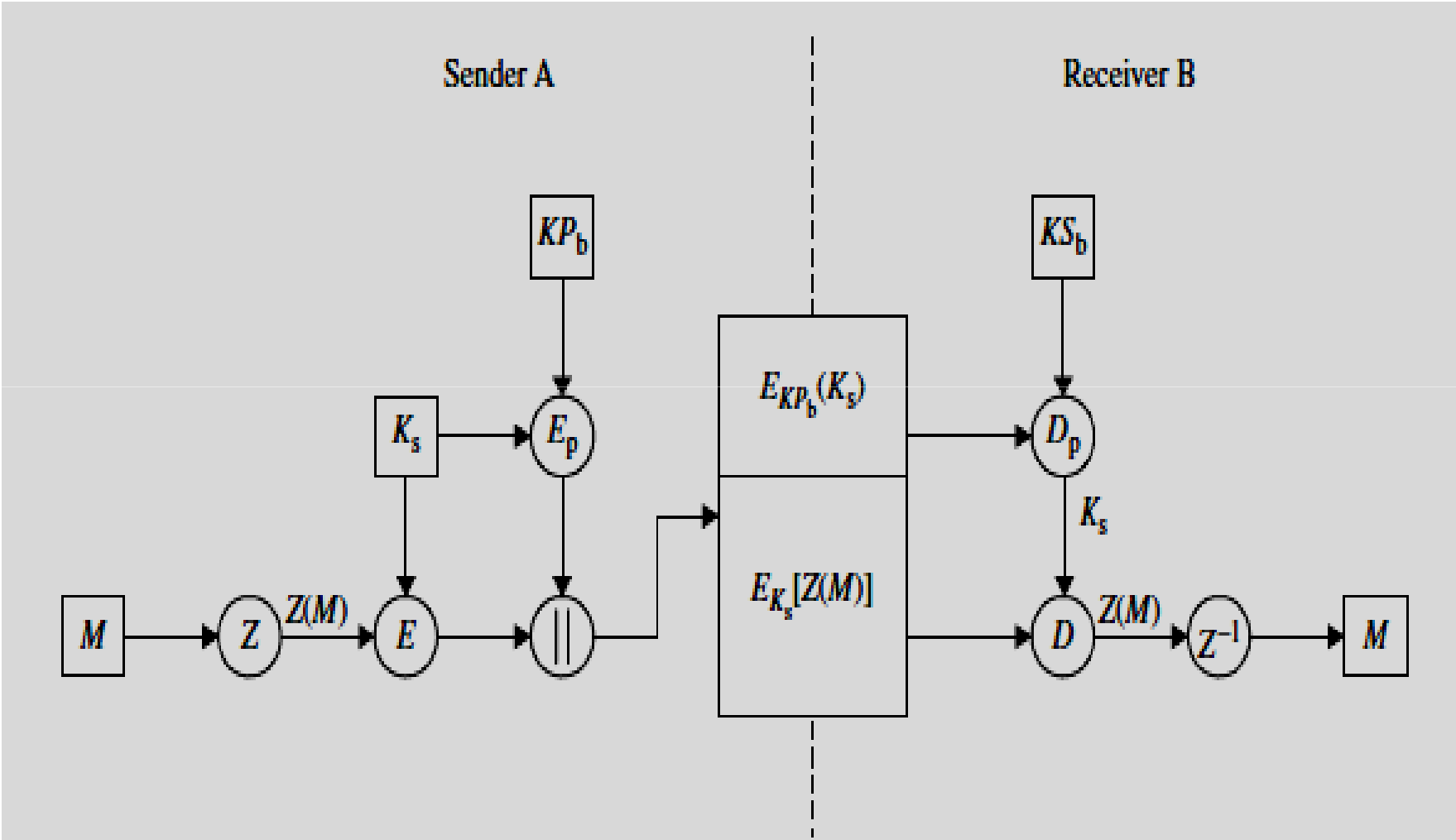


Figure 9.1 PGP confidentiality computation scheme with compression/decompression Algorithms.

PGP

- **Authentication via Digital Signature**

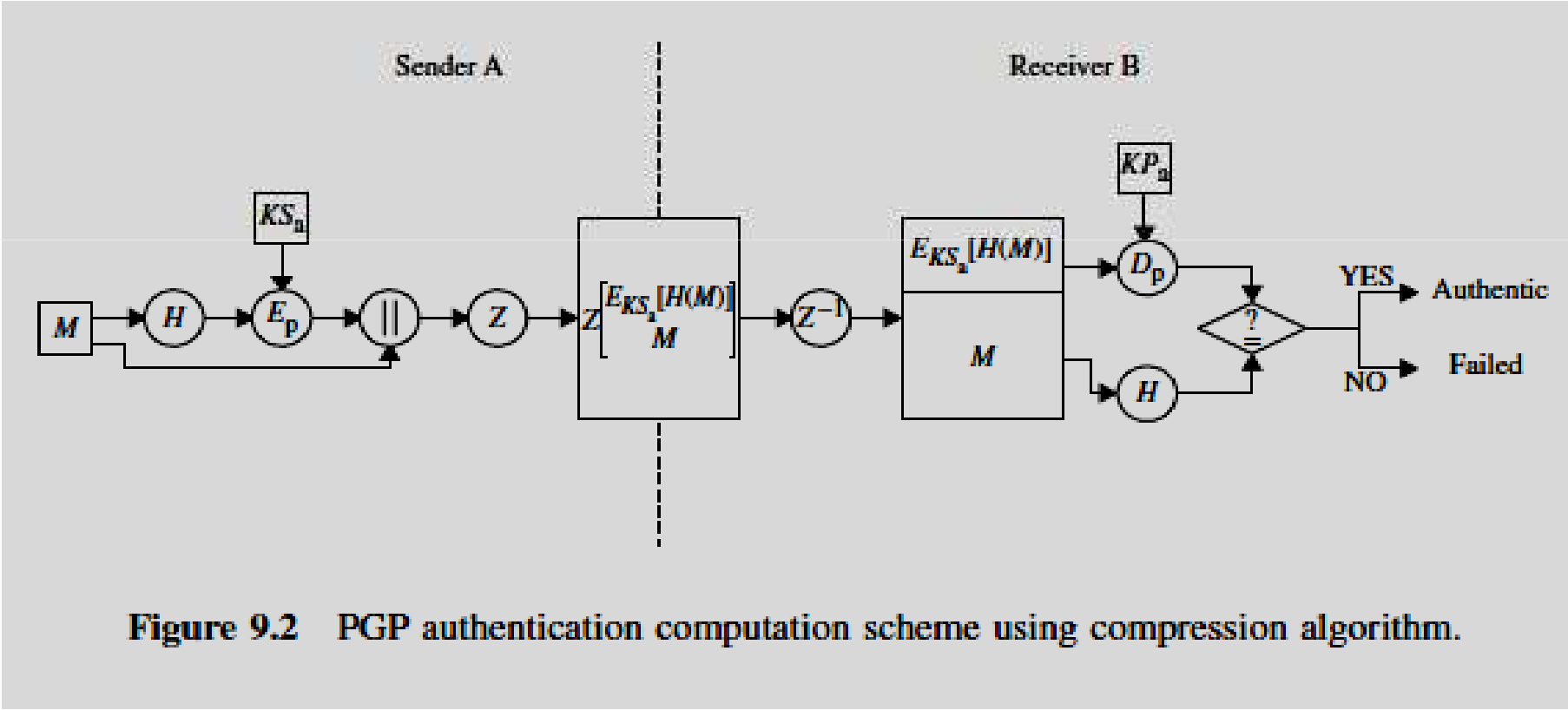


Figure 9.2 PGP authentication computation scheme using compression algorithm.

PGP

- **Compression**
  - *Z compression*
  - *Z*−1 decompression
  - Saving space
    - e-mail transmission
    - file storage
  - Compression will add difficulty- different trade-offs in running speed versus compression ratio produce different compressed forms

PGP

- **Radix-64 Conversion:**

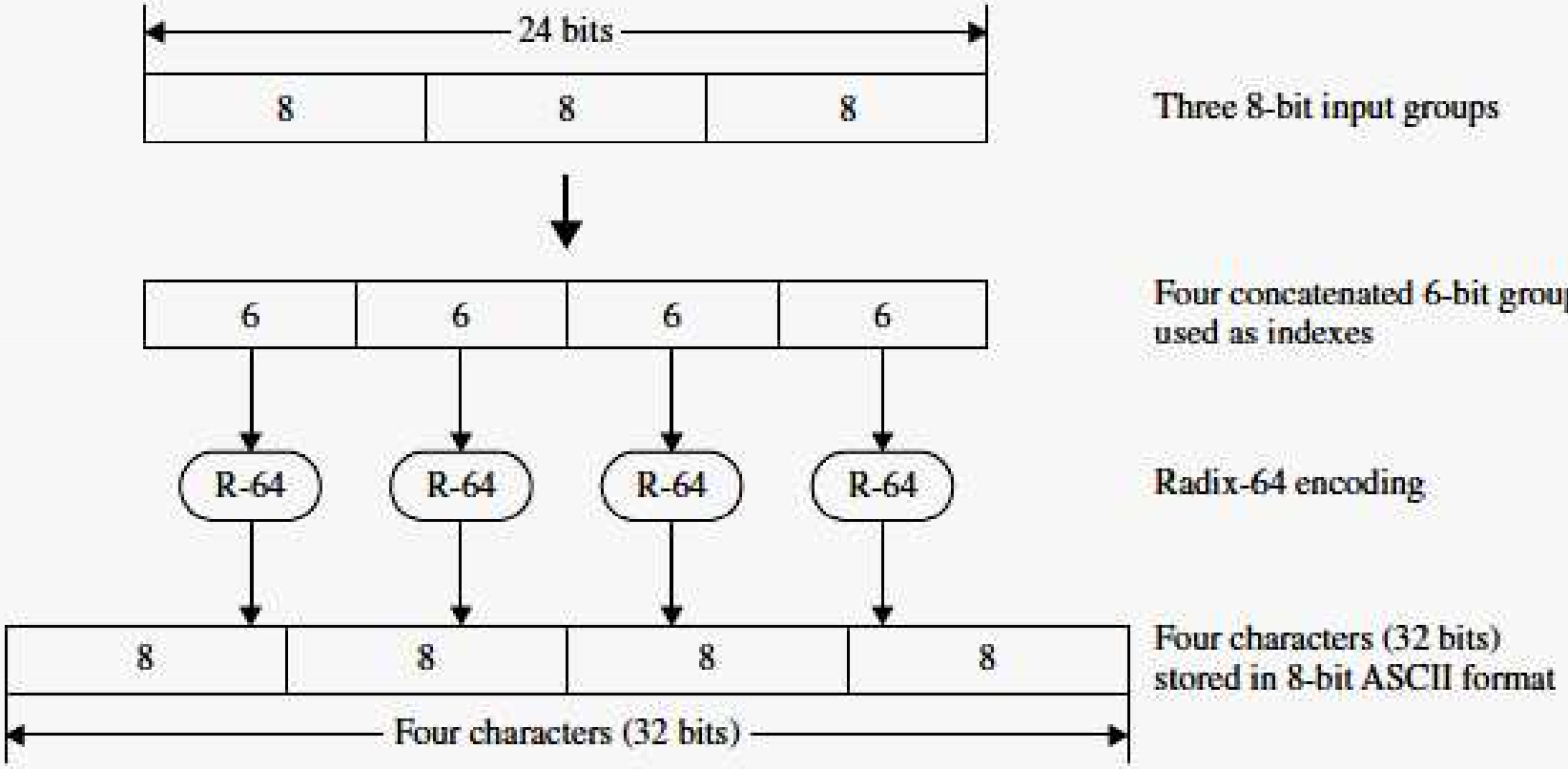


Figure 9.3 Radix-64 printable encoding of binary data.

PGP

- **Radix-64 Conversion**

Table 9.1 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)	=

PGP

ASCII	Hex	Symbol
0	0	NUL
1	1	SOH
2	2	STX
3	3	ETX
4	4	EOT
5	5	ENQ
6	6	ACK
7	7	BEL
8	8	BS
9	9	TAB
10	A	LF
11	B	VT
12	C	FF
13	D	CR
14	E	SO
15	F	SI

ASCII	Hex	Symbol
16	10	DLE
17	11	DC1
18	12	DC2
19	13	DC3
20	14	DC4
21	15	NAK
22	16	SYN
23	17	ETB
24	18	CAN
25	19	EM
26	1A	SUB
27	1B	ESC
28	1C	FS
29	1D	GS
30	1E	RS
31	1F	US

ASCII	Hex	Symbol
32	20	(space)
33	21	!
34	22	"
35	23	#
36	24	$
37	25	%
38	26	&
39	27	'
40	28	(
41	29	)
42	2A	*
43	2B	+
44	2C	,
45	2D	-
46	2E	.
47	2F	/

ASCII	Hex	Symbol
48	30	0
49	31	1
50	32	2
51	33	3
52	34	4
53	35	5
54	36	6
55	37	7
56	38	8
57	39	9
58	3A	:
59	3B	;
60	3C	<
61	3D	=
62	3E	>
63	3F	?

ASCII	Hex	Symbol
64	40	@
65	41	A
66	42	B
67	43	C
68	44	D
69	45	E
70	46	F
71	47	G
72	48	H
73	49	I
74	4A	J
75	4B	K
76	4C	L
77	4D	M
78	4E	N
79	4F	O

ASCII	Hex	Symbol
80	50	P
81	51	Q
82	52	R
83	53	S
84	54	T
85	55	U
86	56	V
87	57	W
88	58	X
89	59	Y
90	5A	Z
91	5B	[
92	5C	\
93	5D	]
94	5E	^
95	5F	_

ASCII	Hex	Symbol
96	60	`
97	61	a
98	62	b
99	63	c
100	64	d
101	65	e
102	66	f
103	67	g
104	68	h
105	69	i
106	6A	j
107	6B	k
108	6C	l
109	6D	m
110	6E	n
111	6F	o

ASCII	Hex	Symbol
112	70	p
113	71	q
114	72	r
115	73	s
116	74	t
117	75	u
118	76	v
119	77	w
120	78	x
121	79	y
122	7A	z
123	7B	{
124	7C	|
125	7D	}
126	7E	~
127	7F	

PGP

- **Radix-64 Conversion**

Raw text	b2	63	29	
24-bit raw text:	10110010	01100011	00101001	
Arranging in blocks of 6 bits	101100	100110	001100	101001
Decimal values are	44,	38,	12,	41
Referring to Table				
Radix-64 encoding	s	m	M	p
ASCII format - hexadecimal	73	6d	4d	70
Binary	0111 0011	0110 1101	0100 1101	0111 0000

PGP

- **Radix-64 Conversion:** *ASCII Armor Format*
- ASCII Armor:
  - Armor head line
  - Armor headers
  - Blank line
  - ASCII-Armored data
  - Armor checksum
  - Armor tail

Example of an ASCII Armored Message

```
-----BEGIN PGP MESSAGE-----
Version: OpenPrivacy 0.99

yDgBO22WxBHv7O8X7O/jygAEzol56iUKiXmV+XmpCtmpqQUKiQrFqclFqUDBovzS
vBSFjNSiVHsuAA==
=njUN
-----END PGP MESSAGE-----
```

PGP

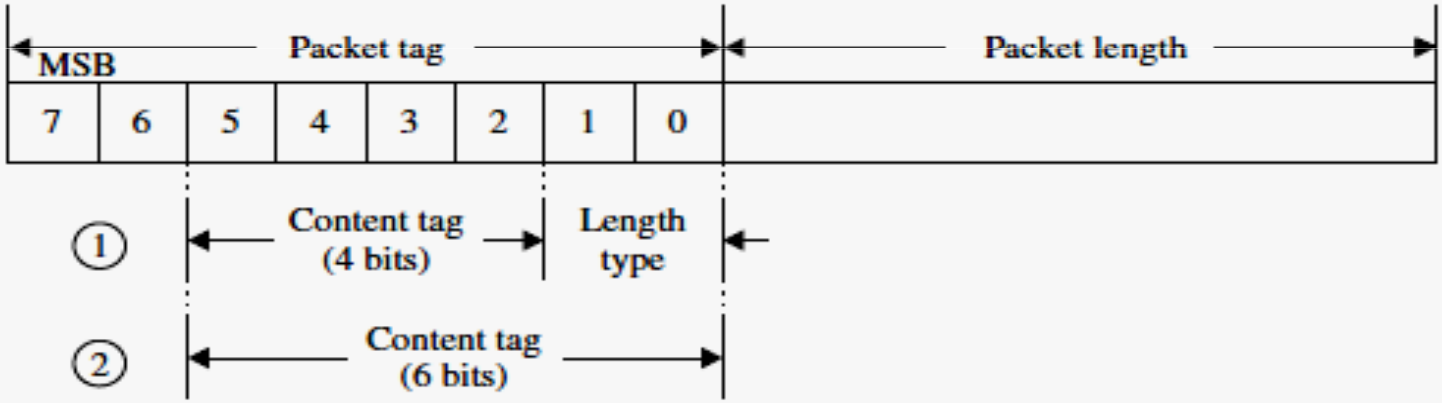
- **Packet**

Packet tag	Packet length	Packet body

Packet header

- **Packet Header**

PGP 2.6.x only uses old format packets



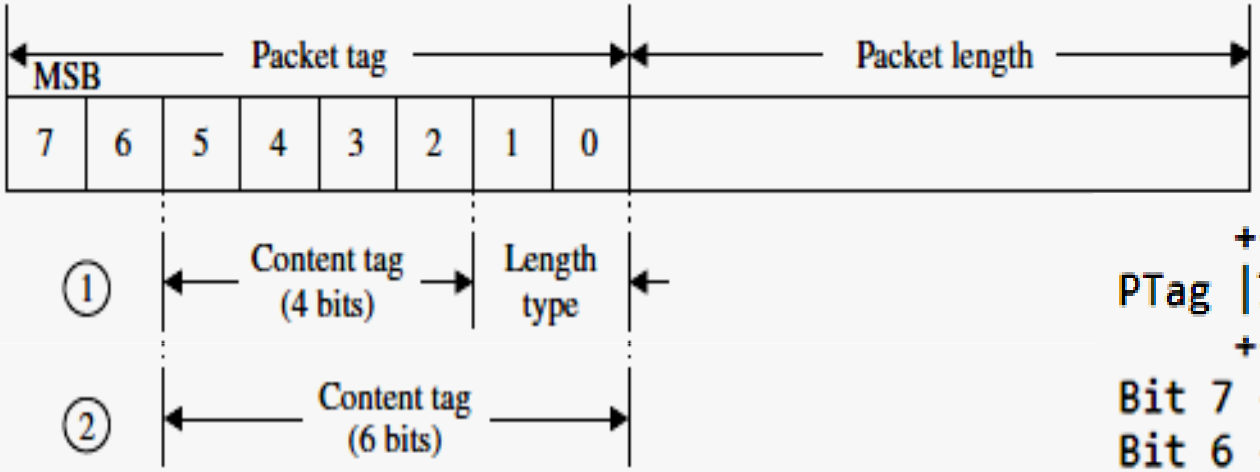
(1) Old format packets: content tag (bits 5, 4, 3, 2); length type (bits 1,0)

(2) New format packets: content tag (bits 5, 4, 3, 2, 1, 0)

Figure 9.4 Packet header.

PGP

- **Packet Header**



1. Old format packets: content tag (bits 5, 4, 3, 2); length type (bits 1,0)

2. New format packets: content tag (bits 5, 4, 3, 2, 1, 0)

Figure 9.4 Packet header.

```
      +---------------+
PTag |7 6 5 4 3 2 1 0|
      +---------------+
Bit 7 -- Always one
Bit 6 -- New packet format if set

Old format packets contain:

      Bits 5-2 -- packet tag
      Bits 1-0 -- length-type

New format packets contain:

      Bits 5-0 -- packet tag
```

PGP

- **Packet Header**
  - **Example**

1. Packet data length 100 *(decimal) = 01100100(binary) = 0x64 (hex)*
   - *Need* one octet
   - So packet header length is octet long
   - This header is followed by 100 octets of data
2. Length 1723
   - two octets: 0xc5, 0xfb
   - This header is followed by the 1723 octets of data
3. Length 100000
   - five octets: 0xff, 0x00, 0x01, 0x86, 0xa0.

PGP

- **PGP Packet Structure**
- A PGP file
  - Concatenation of one or more packets
  - Packets in a file may be subject to a transformation using
    - Encryption
    - Compression
    - Digital signature
    - Radix-64 conversion

PGP

- **PGP Packet Structure**
  - Message packet Signature packet Session key packet

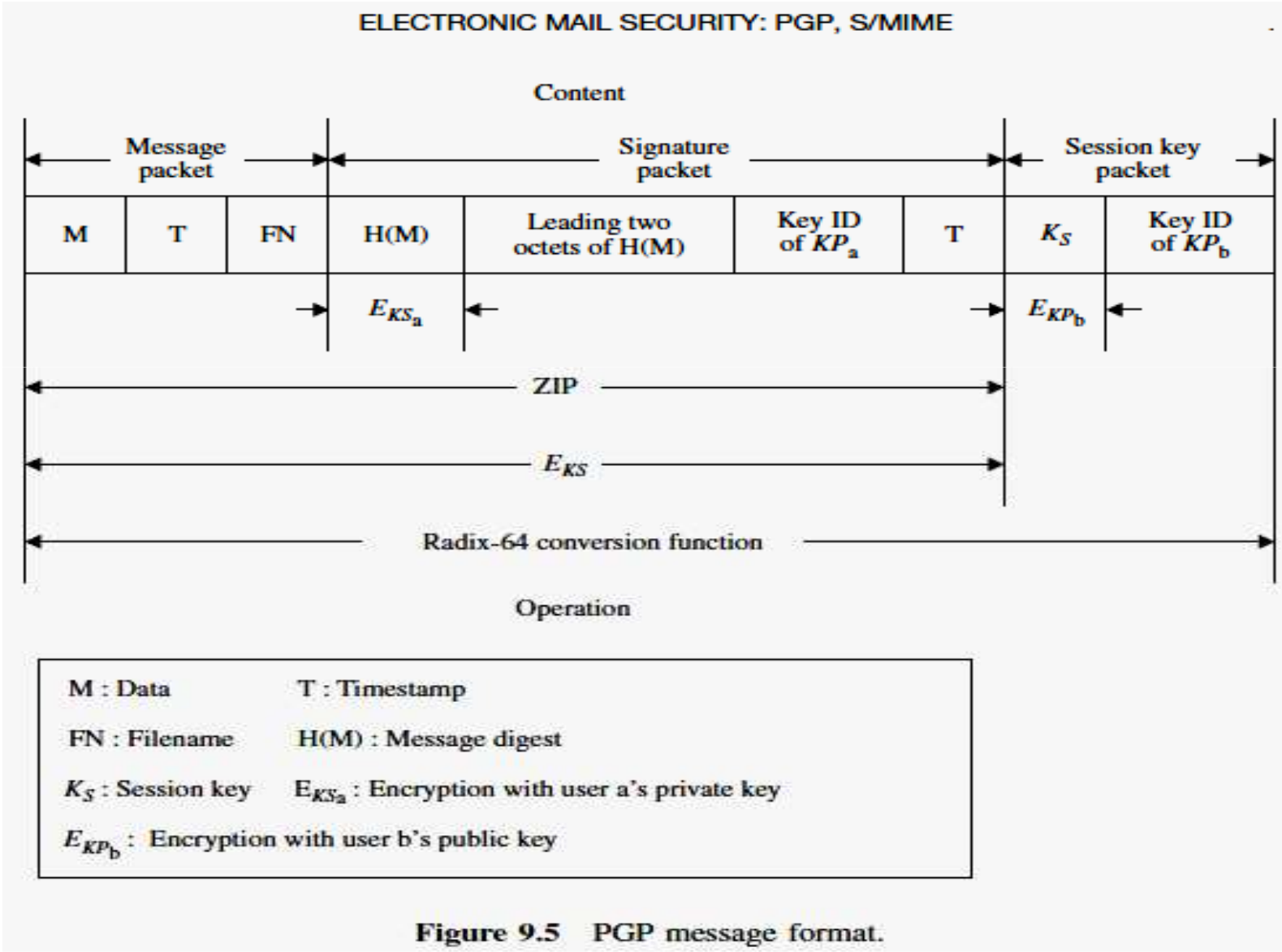


Figure 9.5 PGP message format.

PGP

- **PGP Packet Structure**
- Message packet
  - Contains
    - Actual data
    - Filename
    - Timestamp

PGP

- **PGP Packet Structure**
- *Signature Packet (Tag 2)*
  - Describes a binding between some public key and some data
  - Components
    - Timestamp
    - Message digest
    - Leading two octets of hash code
    - Key ID of sender's public key

PGP

- ***PGP Packet Structure***
- Session Key Packets (Tag 1)
  - (E_KPb (Ks) || ID of receiver's public key)
  - The body of this session key component consists of:
    - A one-octet version number which is 3
    - An eight-octet - key ID of the public key
    - A one-octet number giving the public key algorithm used
    - A string of octets that is the encrypted session key

```
New: Public-Key Encrypted Session Key Packet(tag 1)(524 bytes)
    New version(3)
    Key ID - 0x2478DF98CED355D3
    Pub alg - RSA Encrypt or Sign(pub 1)
    RSA m^e mod n(4096 bits) - ...
        -> m = sym alg(1 byte) + checksum(2 bytes) + PKCS-1 block type 02
```

PGP

- **Key Material Packet**
  - Contains all the information about a public or private key
  - 4 variants
  - 2 versions : version 3 and version 4
  - *Key Packet Variants*
    - **Public-Key Packet (Tag 6)**
    - **Public-Subkey Packet (Tag 14)**
    - **Secret-Key Packet (Tag 5)**
    - **Secret-Subkey Packet (Tag 7)**

PGP

- **Key Material Packet**
  - **Public-key packet (tag 6):**
    - This packet starts a series of packets that forms a PGP 5.x key
  - **Public subkey packet (tag 14):**
    - This packet has exactly the same format as a publickey packet
    - Denotes a subkey
    - One or more subkeys may be associated with atop-level key
    - The top-level key provides signature services
    - The subkeys provide encryption services
  - **Secret-key packet (tag 5):**
    - This packet contains all the information that is found in a public-key packet
    - Also includes the secret-key material after all the public-key fields
  - **Secret-subkey packet (tag 7):**
    - It is the subkey analogous to the secret-key packet

PGP

- **Key Material Packet**
- *Public-key Packet Formats*
  - Version 2,3,4
  - A v3 key packet contains:
    - A one-octet version number (3)
    - A four-octet number denoting the time that the key was created
    - A two-octet number denoting the time in days that this key is valid
    - A one-octet number denoting the public-key algorithm of this key
    - A series of multiprecision integers (MPIs) comprising the key material: an MPI of RSA public module *n; an MPI of RSA public encryption exponent e*

PGP

- **Key Material Packet**
- *Public-key Packet Formats*
  - A v4 key packet contains:
    - A one-octet version number (4)
    - A four-octet number denoting the time that the key was created
    - A one-octet number denoting the public-key algorithm of this key
    - A series of MPIs comprising the key material

```
New: Public Key Packet(tag 6)(269 bytes)
    Ver 4 - new
    Public key creation time - Mon May 25 15:12:34 PDT 2015
    Pub alg - RSA Encrypt or Sign(pub 1)
    RSA n(2048 bits) - ...
    RSA e(17 bits) - ...
```

3.6.1 String-to-key (S2k) specifier types

There are three types of S2K specifiers currently supported, as follows:

3.6.1.1 Simple S2K

This directly hashes the string to produce the key data. See below for how this hashing is done.

```
    Octet 0:        0x00
    Octet 1:        hash algorithm
```

3.6.1.2 Salted S2K

This includes a "salt" value in the S2K specifier -- some arbitrary data -- that gets hashed along with the passphrase string, to help prevent dictionary attacks.

```
    Octet 0:        0x01
    Octet 1:        hash algorithm
    Octets 2-9:     8-octet salt value
```

3.6.1.3 Iterated and Salted S2K

This includes both a salt and an octet count. The salt is combined with the passphrase and the resulting value is hashed repeatedly. This further increases the amount of work an attacker must do to try dictionary attacks.

Octet 0:	0x04
Octet 1:	hash algorithm
Octets 2-9:	8-octet salt value
Octets 10-13:	count, a four-octet, unsigned value

PGP

- **Algorithms for PGP 5.x**

Public-Key Algorithms

ID	Algorithm
1	RSA (encrypt or sign)
2	RSA encryption only
3	RSA sign only
16	ElGamal (encrypt only)
17	DSA (DSS)
18	Reserved for elliptic curve
19	Reserved for ECDSA
20	ElGamal (encrypt or sign)
21	Reserved for Diffie–Hellman
100–110	Private/experimental algorithm

PGP

- **Algorithms for PGP 5.x**

Symmetric-Key Algorithms

ID	Algorithm
0	Plaintext or unencrypted data
1	IDEA
2	Triple DES (DES–EDE)
3	CAST 5 (128-bit key)
4	Blowfish (128-bit key, 16 rounds)
5	SAFER-SK128 (13 rounds)
6	Reserved for DES/SK
7	Reserved for AES (128-bit key)
8	Reserved for AES (192-bit key)
9	Reserved for ASE (256-bit key)
100–110	Private/experimental algorithm

PGP

- **Algorithms for PGP 5.x**

Compression Algorithm

ID	Algorithm
0	Uncompressed
1	ZIP (RFC 1951)
2	ZLIB (RFC 1950)
100–110	Private/experimental algorithm

PGP

- **Algorithms for PGP 5.x**

Hash Algorithms

ID	Algorithm
1	MD5
2	SHA-1
3	RIPE-MD/160
4	Reserved for double-width SAH (experimental)
5	MD2
6	Reserved for TIGER/192
7	Reserved for HAVAL (5 pass, 160-bit)
100–110	Private/experimental algorithm