Security at Application Layer (PGP,S/MIME)

Module 5.3

Why Study E-mail Security?

- After web browsing, e-mail is the most widely used network-reliant application.
- Mail servers, after web servers, are the most often attacked Internet hosts.
- Basic e-mail offers little security, counter to public perception.
- Good technical solutions are available, but not widely used.
    - If we understand why this is so, we might understand something about why security is ‘hard’.

Threats to E-mail

- Loss of confidentiality.
  - *E-mails are sent in clear over open networks.*
  - *E-mails stored on potentially insecure clients and mail servers.*
- Loss of integrity.
  - *No integrity protection on e-mails; anybody be altered in transit or on mail server.*
- Lack of data origin authentication.
  - *Is this e-mail really from the person named in the* From:*field?*

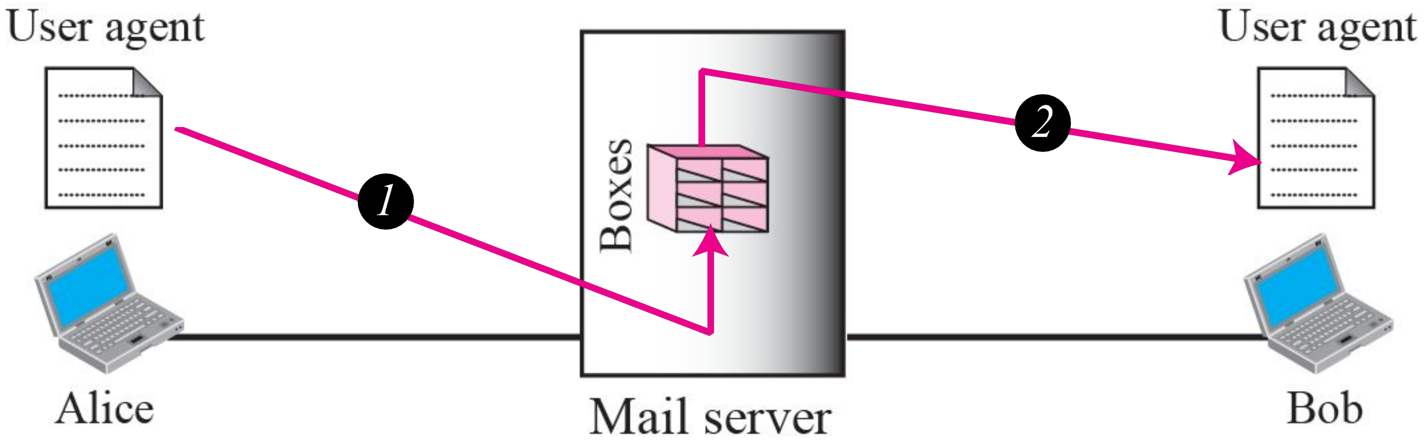
Threats to E-mail

- Lack of non-repudiation.
  - *Can I rely and act on the content? (integrity)*
  - *If so, can the sender later deny having sent it? Who is liable if I have acted?*
- Lack of notification of receipt.
  - *Has the intended recipient received my e-mail and acted on it?*
  - *A message locally marked as 'sent' may not have been delivered.*

E-mail security

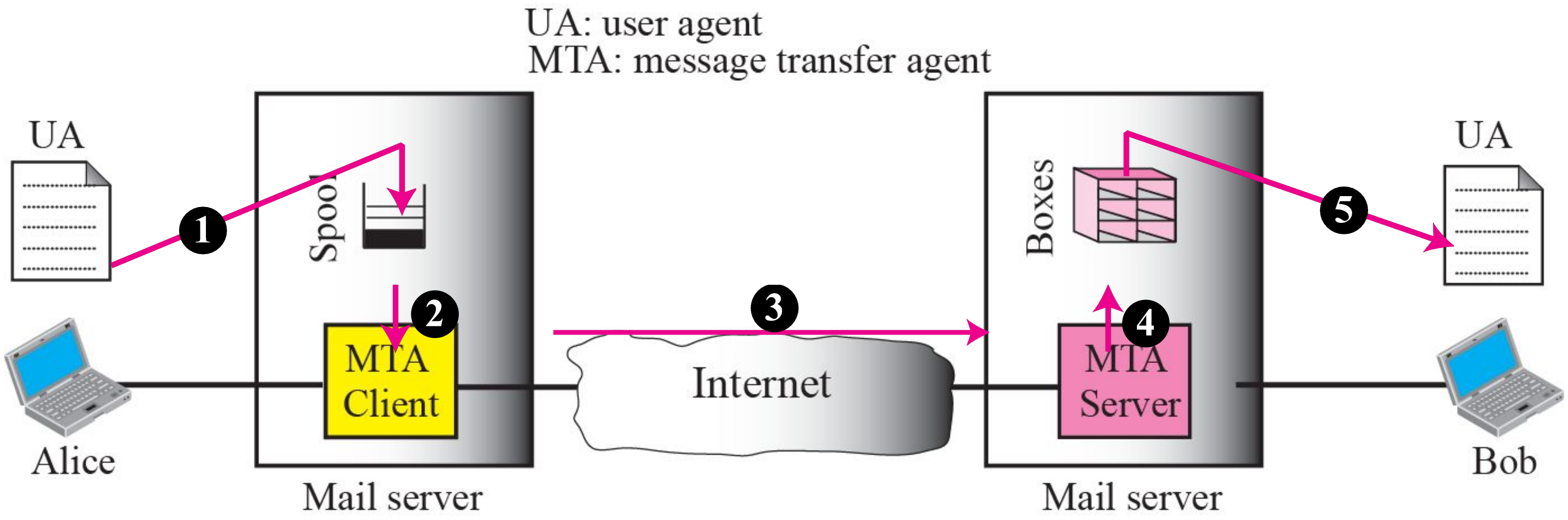
- What are the Options?
  - Secure the server to client connections (easy thing first)
    - https access to webmail
    - Protection against insecure wireless access
  - Secure the end-to-end email delivery
    - The PGPs of the world
    - Practical in an enterprise intra-network environment

Scenario I



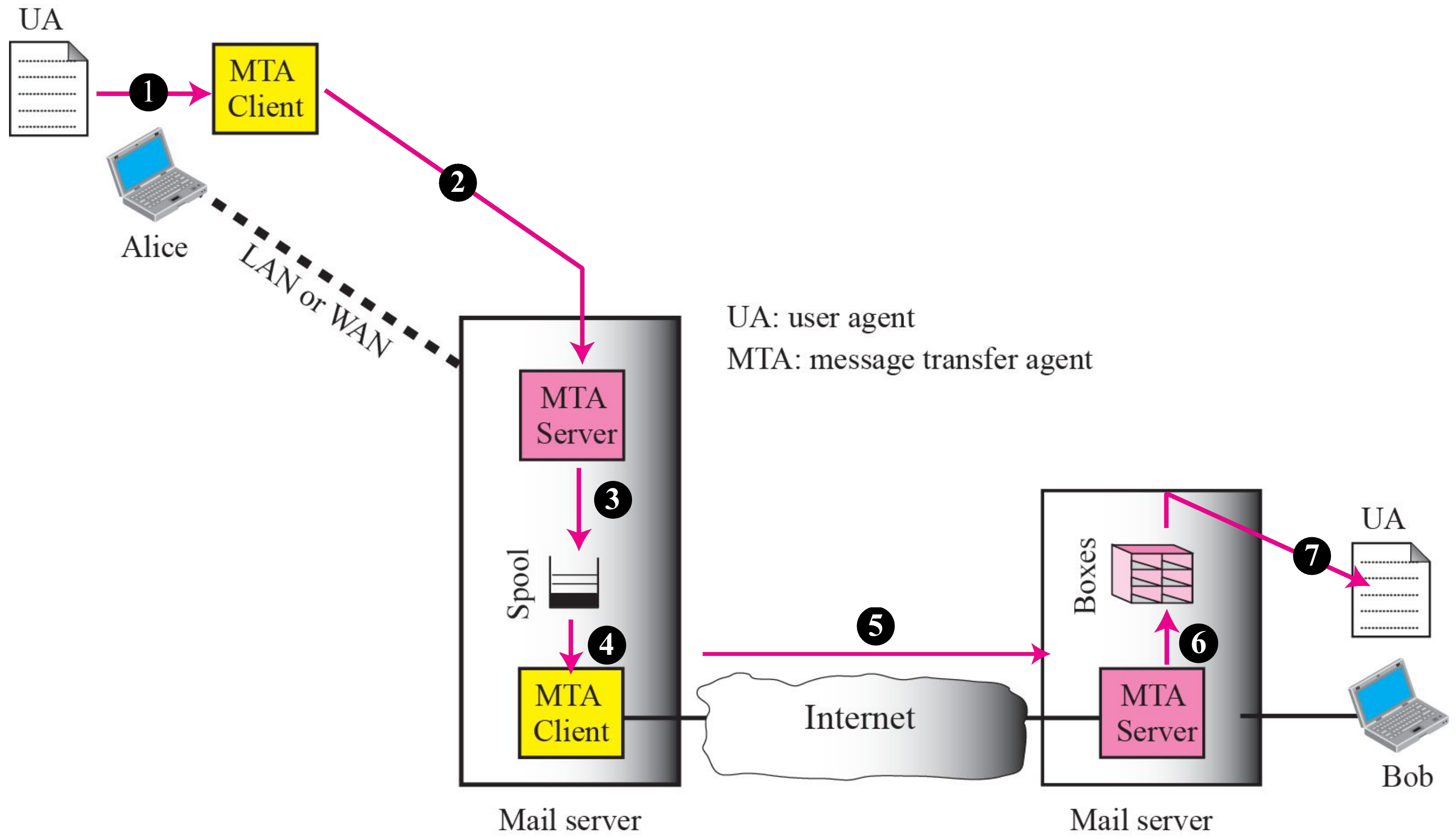
When the sender and the receiver of an e-mail are on the same mail server, we need only two user agents.

Scenario II



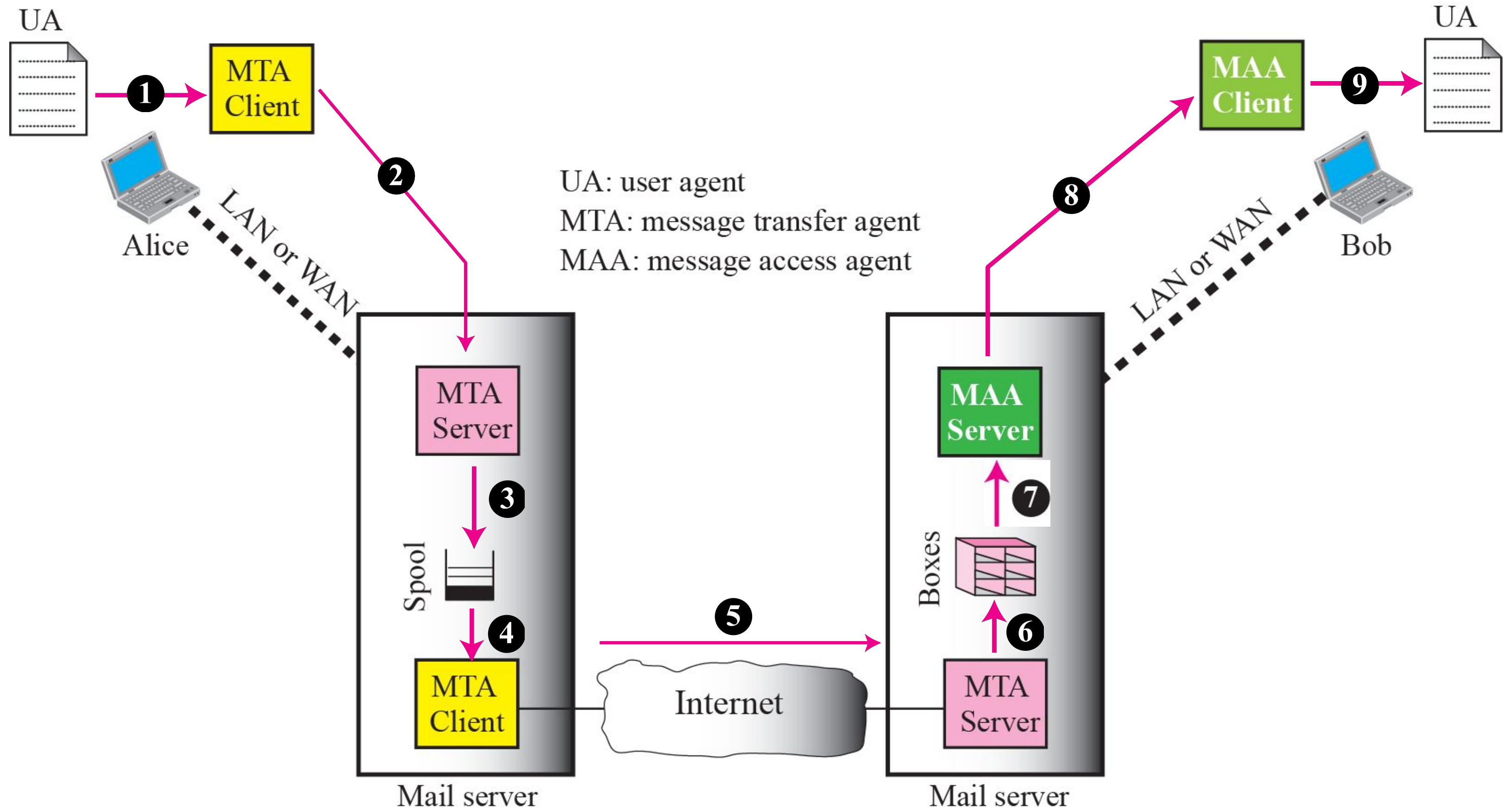
When the sender and the receiver of an e-mail are on different mail servers,
we need two UAs and a pair of MTAs (client and server).

Scenario III



When the sender is connected to the mail server via a LAN or a WAN, we need two UAs and two pairs of MTAs (client and server).

Scenario IV



When both sender and receiver are connected to the mail server via a LAN or a WAN, we need two UAs, two pairs of MTAs (client and server), and a pair of MAAs (client and server). This is the most common situation today.

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

PGP and S/MIME

- S/MIME and PGP are both protocols used for authentication and privacy to messages over the internet.
- PGP, stands for Pretty Good Privacy, is a data encryption and decryption computer program that offers cryptographic privacy and authentication for Internet data transmission.
- PGP is widely used for signing, encrypting and decrypting electronic data to maximize the security issues of data exchange
- The protocol S/MIME refers to Secure/Multipurpose Internet Mail Extensions.
- S/MIME is recently included in the latest versions of the web browsers from renowned software companies like Microsoft and Netscape and has also been broadly accepted by many vendors in all around the world. It is also driven as a standard for public key encryption and signing of MIME data. S/MIME is based on an IETF standard and most commonly defined in RFCs documents.
- S/MIME provides the authentication, message integrity and non-repudiation of origin and data security services for electronic data transmission applications.
- S/MIME is very closely similar to PGP and its predecessors.and usesX.509v3 format for certificates.

PGP and S/MIME

- PGP encryption uses a serial combination of hashing, data compression, symmetric-key cryptography, and public-key cryptography.
- While using PGP, one user has the ability to give directly a public key to another user or the second user can obtain the public key from the first user.
- PGP does not mandate a policy for creating trust and hence each user is free to decide the length of trust in the received keys. With the S/MIME, the sender or receiver does not rely on exchanging keys in advance and share a common certifier on which both can rely.
- S/MIME is considered superior to PGP from an administrative perspective because of its strength, support for centralized key management through X.509 certificate servers
- S/MIME protocol allows most vendors to send and receive encrypted email without using additional software.
- S/MIME is convenient because of secure transformation of all applications like spreadsheets, graphics, presentations, movies etc., but PGP was originated to address the security concerns of plain e-mail or text messages. S/MIME is also highly affordable in terms of its cost.

PGP and S/MIME difference

- S/MIME and PGP protocols use different formats for key exchange.
- PGP depends upon each user's key exchange S/MIME uses hierarchically validated certifier for key exchange. PGP was developed to address the security issues of plain text messages.
- But S/MIME is designed to secure all kinds of attachments/data files.
- S/MIME is known to dominate the secure electronic industry because it is incorporated into many commercial e-mail packages.
- S/MIME products are cheaply available than for PGP.

PGP

- PGP is an open-source, freely available software package for e-mail security.
- It provides authentication through the use of digital signature, confidentiality through the use of symmetric block encryption, compression using the ZIP algorithm, and e-mail compatibility using the radix-64 encoding scheme.
- PGP incorporates tools for developing a public-key trust model and public-key certificate management.
- S/MIME is an Internet standard approach to e-mail security that incorporates the same functionality as PGP.

Pretty Good Privacy (PGP)

- widely used de facto secure email
- developed by Phil Zimmermann
- general purpose application to protect (encrypt and/or sign) files
- can be used by corporations as well as individuals
- selected best available crypto algs to use(IDEA, RSA, SHA-1)
- integrated into a single program
- available on Unix, PC, Macintosh and Amiga systems
- originally free, now have commercial versions available also(http://www.pgpi.org)
- PGP is now on an Internet standards track (RFC 3156)

PGP- Services

- There are four services included in the operation of PGP
  - Authentication
  - Confidentiality
  - Compression
  - E-mail compatibility
  - Segmentation and Reassembly

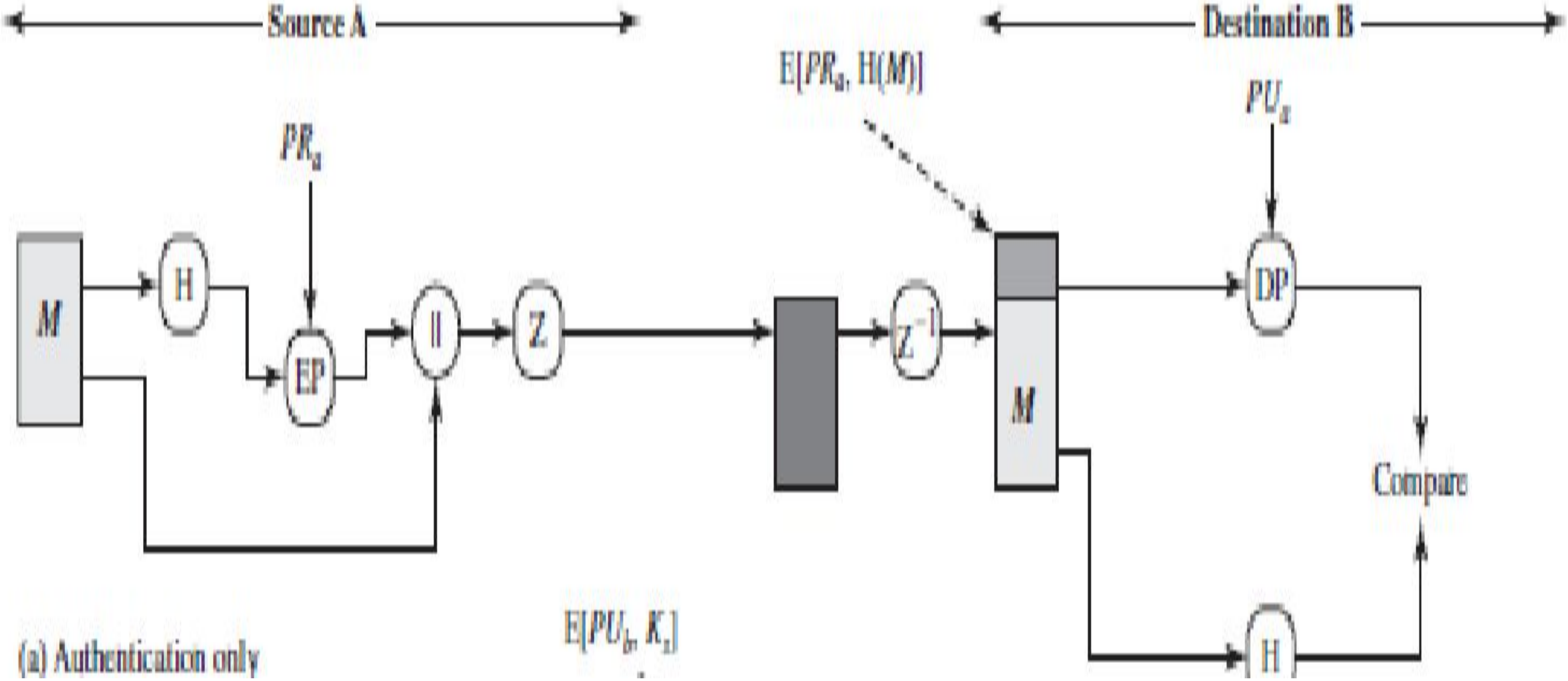
Conventions used

- K_s = session key used in symmetric encryption scheme
- Pr_a =private key of user A, used in public-key encryption scheme
- PU_a = public key of user A, used in public-key encryption scheme
- EP = public-key encryption
- DP = public-key decryption
- EC = symmetric encryption
- DC = symmetric decryption
- H = Hash function
- || = concatenation
- Z = compression using Zip algorithm
- R64 = Radix format

PGP Operation – Authentication

1. sender creates a message
2. SHA-1 used to generate 160-bit hash code of message
3. hash code is encrypted with RSA using the sender's private key, and result is attached to message
4. receiver uses RSA or DSS with sender's public key to decrypt and recover hash code
5. receiver generates new hash code for message and compares with decrypted hash code, if match, message is accepted as authentic

Authenticate

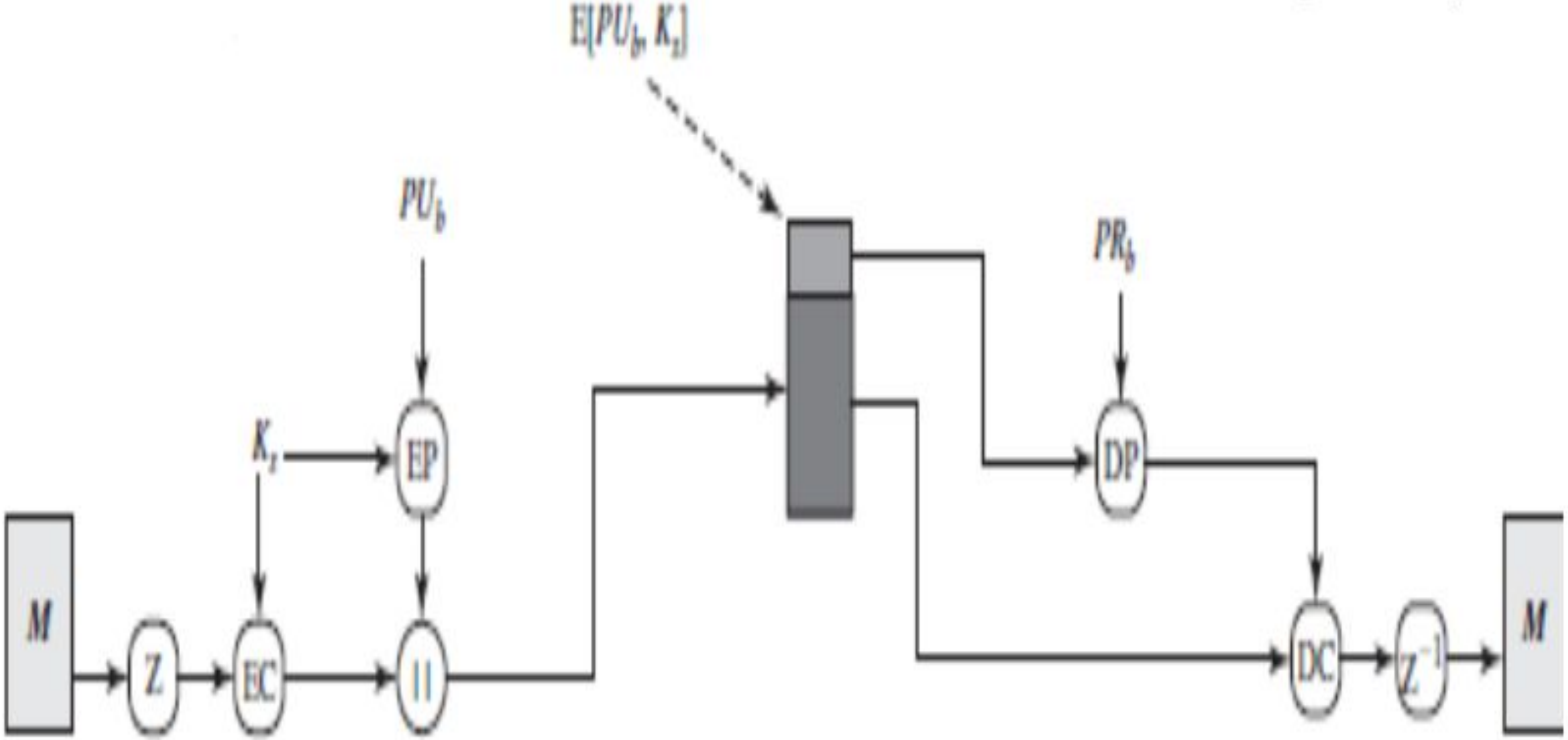


(a) Authentication only

PGP Services – Confidentiality

1. sender generates message and random 128-bit number to be used as session key for this message only
2. message is encrypted, using CAST-128 / IDEA/3DES with session key
3. session key is encrypted using RSA with recipient's public key, then attached to message
4. receiver uses RSA with its private key to decrypt and recover session key
5. session key is used to decrypt message

confidentiality

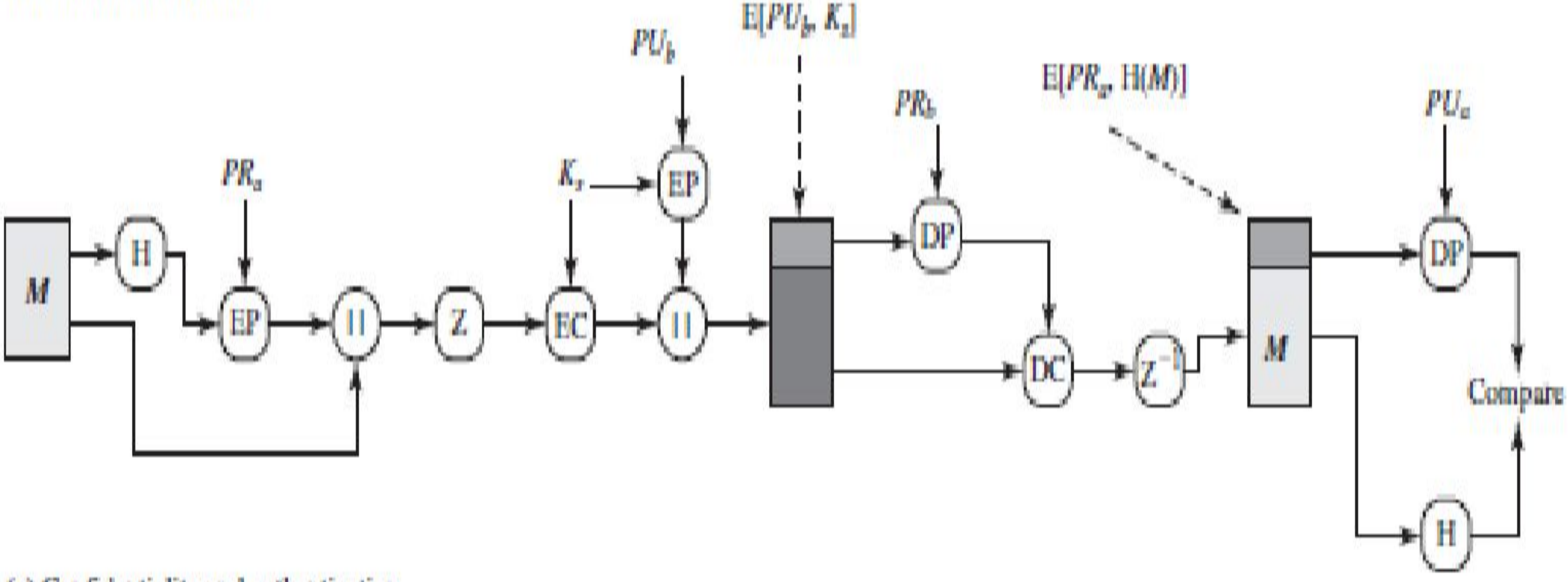


(b) Confidentiality only

PGP Operation – Confidentiality & Authentication

- uses both services on same message
  - create signature & attach to message
  - encrypt both message & signature
  - attach RSA encrypted session key

Confidentiality and Authentication



(c) Confidentiality and authentication

PGP Services – Compression

- By default PGP compresses message after signing but before encrypting
  - so can store uncompressed message & signature for later verification
  - & because compression is non deterministic
- uses ZIP compression algorithm

PGP Services – Compression

The signature is generated before compression for two reasons:

- It is preferable to sign an uncompressed message so that one can store only the uncompressed message together with the signature for future verification . If one signed a compressed document, then it would be necessary either to store a compressed version of the message for later verification or to recompress the message when verification is required.
- PGP’s compression algorithm presents a difficulty. The algorithm is not deterministic; various implementations of the algorithm achieve different tradeoffs in running speed versus compression ratio different compressed forms.

PGP Operation – Compression

- Message encryption is applied after compression to strengthen cryptographic security.
- The compressed message has less redundancy than the original plaintext, cryptanalysis is more difficult.
- The compression algorithm used is ZIP

PGP Operation – Email Compatibility

- when using PGP will have binary data to send (encrypted message etc)
- however email was designed only for text
- hence PGP must encode raw binary data into printable ASCII characters
- uses radix-64 algorithm
  - maps 3 bytes to 4 printable chars
  - also appends a CRC
- PGP also segments messages if too big

PGP E-Mail Compatibility

- Many electronic mail systems can only transmit blocks of ASCII text.
- This can cause a problem when sending encrypted data since ciphertext blocks might not correspond to ASCII characters which can be transmitted.
- PGP overcomes this problem by using radix-64 conversion.

Radix-64 Conversion

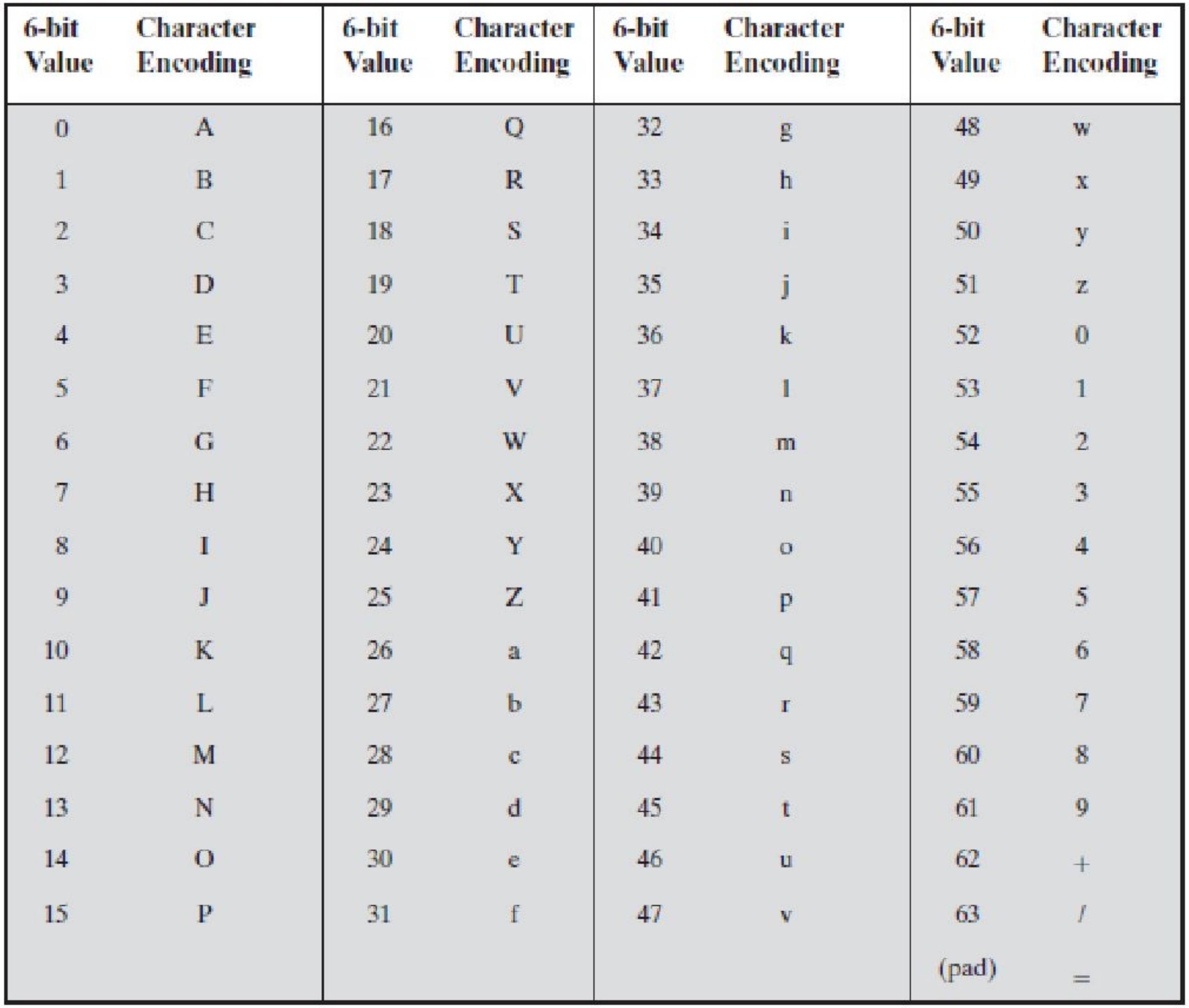
1. The binary input is split into blocks of 24 bits (3 bytes).
2. Each 24 block is then split into four sets each of 6-bits.
3. Each 6-bit set will then have a value between 0 and 2^6-1 (=63).
4. This value is encoded into a printable character.

Radix-64 Conversion: Example

- Suppose the email message is: new
- ASCII format: 01101110 01100101 01110111
- After encryption: 10010001 10011010 10001000
- The Radix-64 conversion:
  - The 24-bit block: 10010001 10011010 10001000
  - Four 6-bit blocks: 100100 011001 101010 001000
  - Integer version: 36 25 38 8
  - Printable version: k Z m I

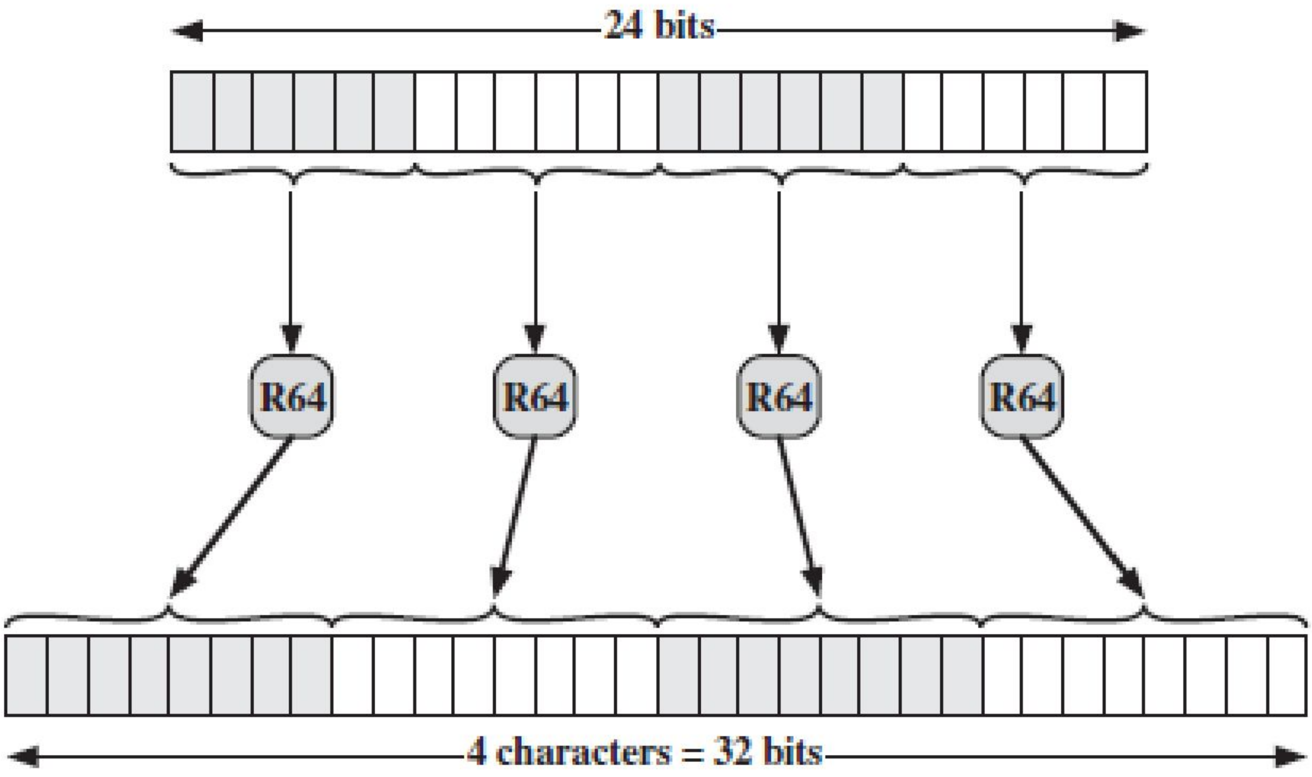
Radix 64 conversion

Table 18.9 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)	=

Radix conversion

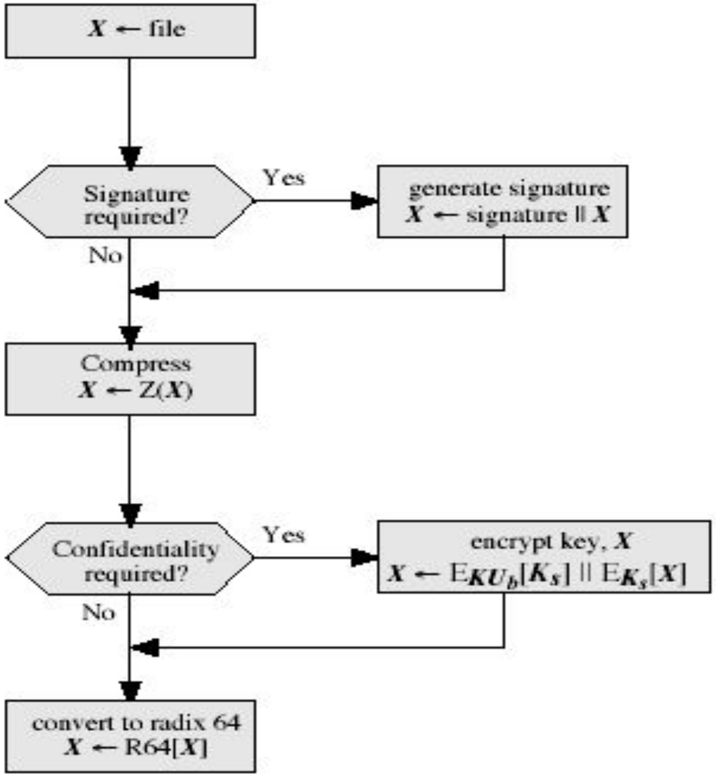


Use of radix conversion expands the message by 33%

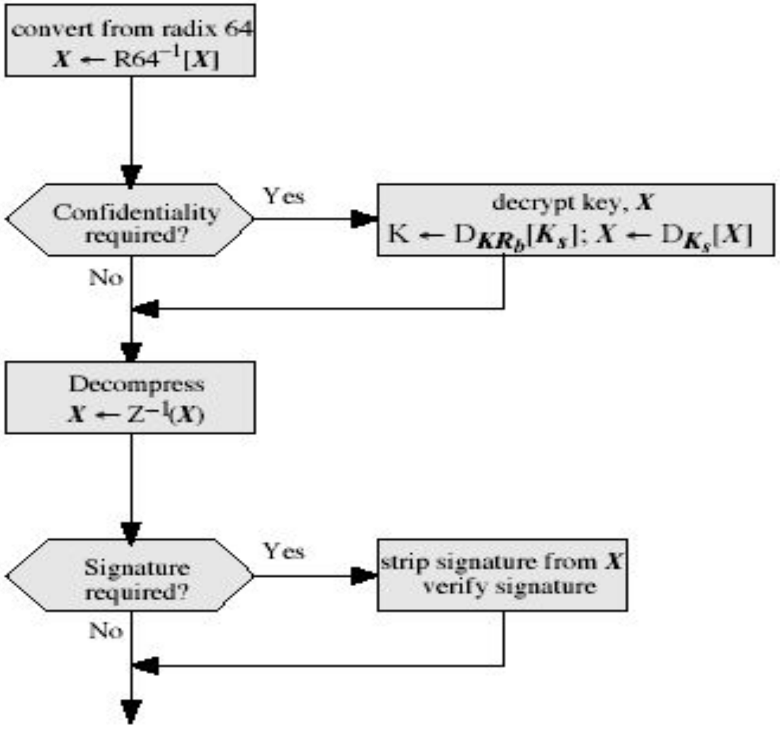
PGP Operation - Segmentation/Reassembly

- E-mail facilities often are restricted to a maximum message length.
- For example, many of the facilities accessible through the Internet impose a maximum length of 50,000 octets.
- Any message longer than that must be broken up into smaller segments, each of which is mailed separately.
- To accommodate this restriction, PGP automatically subdivides a message that is too large into segments that are small enough to send via e-mail.
- The segmentation is done after all of the other processing, including the radix-64 conversion. Thus, the session key component and signature component appear only once, at the beginning of the first segment. Reassembly at the receiving end is required before verifying signature or decryption

PGP Operation – Summary

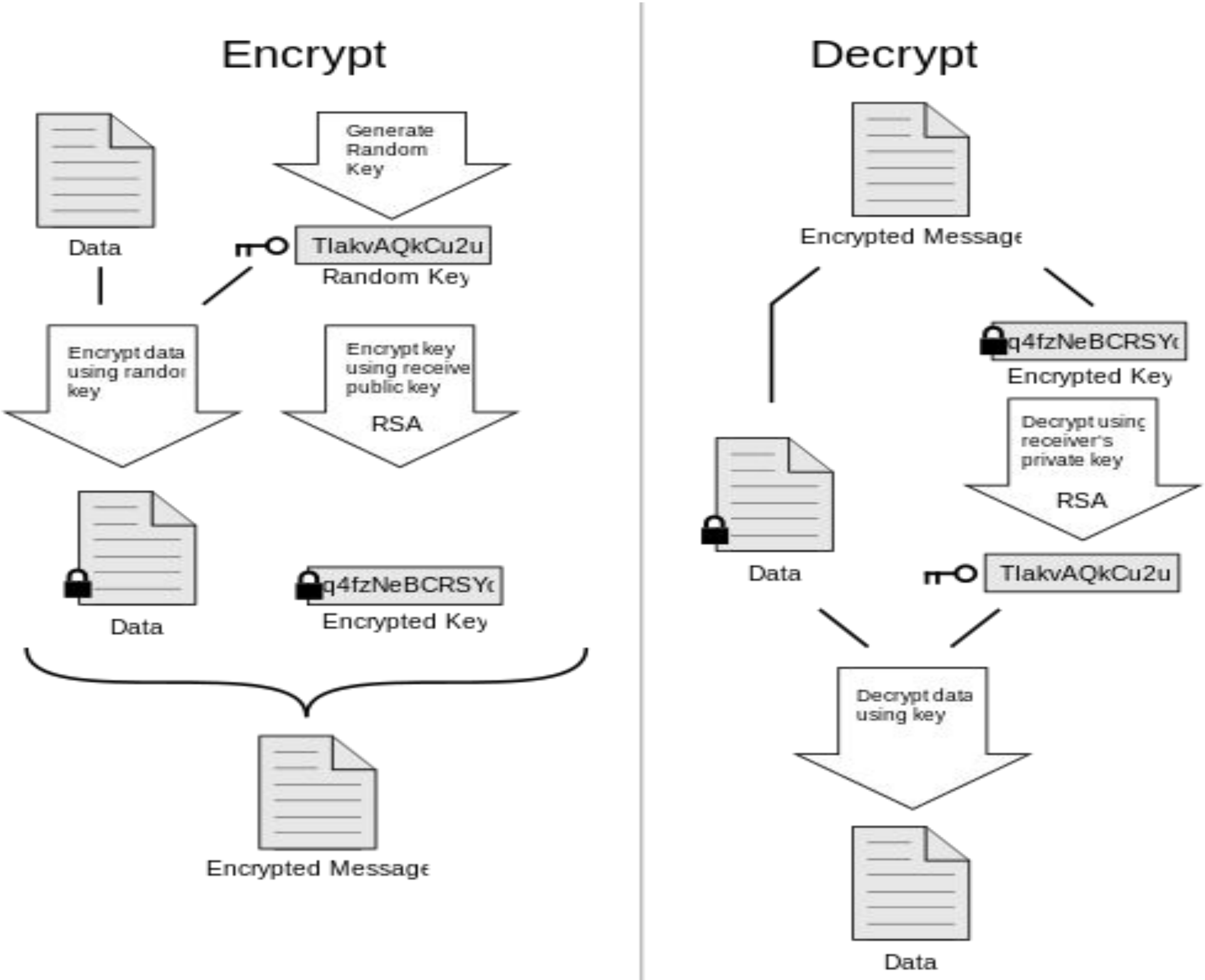


(a) Generic Transmission Diagram (from A)



(b) Generic Reception Diagram (to B)

PGP Process


Encrypt
Generate Random Key
Data
TlakvAQkCu2u
Random Key
Encrypt data using randor key
Encrypt key using receive public key
RSA
Data
q4fzNeBCRSY‹
Encrypted Key
Encrypted Message
Decrypt
Encrypted Message
q4fzNeBCRSY‹
Encrypted Key
Decrypt using receiver's private key
RSA
Data
TlakvAQkCu2u
Decrypt data using key
Data

PGP Keys

- Makes use of four type of keys
  - One time session conventional keys
    - Associated with a single message
  - Public keys
    - used in asymmetric encryption
  - Private keys
    - Also used in asymmetric encryption
  - Public / private key pairs are the most expensive to generate.
  - A single user can have multiple public/private key pairs.
  - Since the security of the system depends on protecting private keys, these are encrypted using a passphrase system.
  - Passphrase based conventional keys
    - used to protect private keys

PGP Session Keys

- Each session key is associated with a single message and is used only for the purpose of encryption and decrypting that message.
- 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
- The input to the random number generator consists of as 128-bit key (this is a random number using the keystroke input from the user) and two 64-bit blocks that are treated as plaintext to be encrypted
- The encryption algorithm E is used to generate a new n-bit key from a previous session key and two n /2-bit blocks generated based on user keystrokes, including keystroke timing.

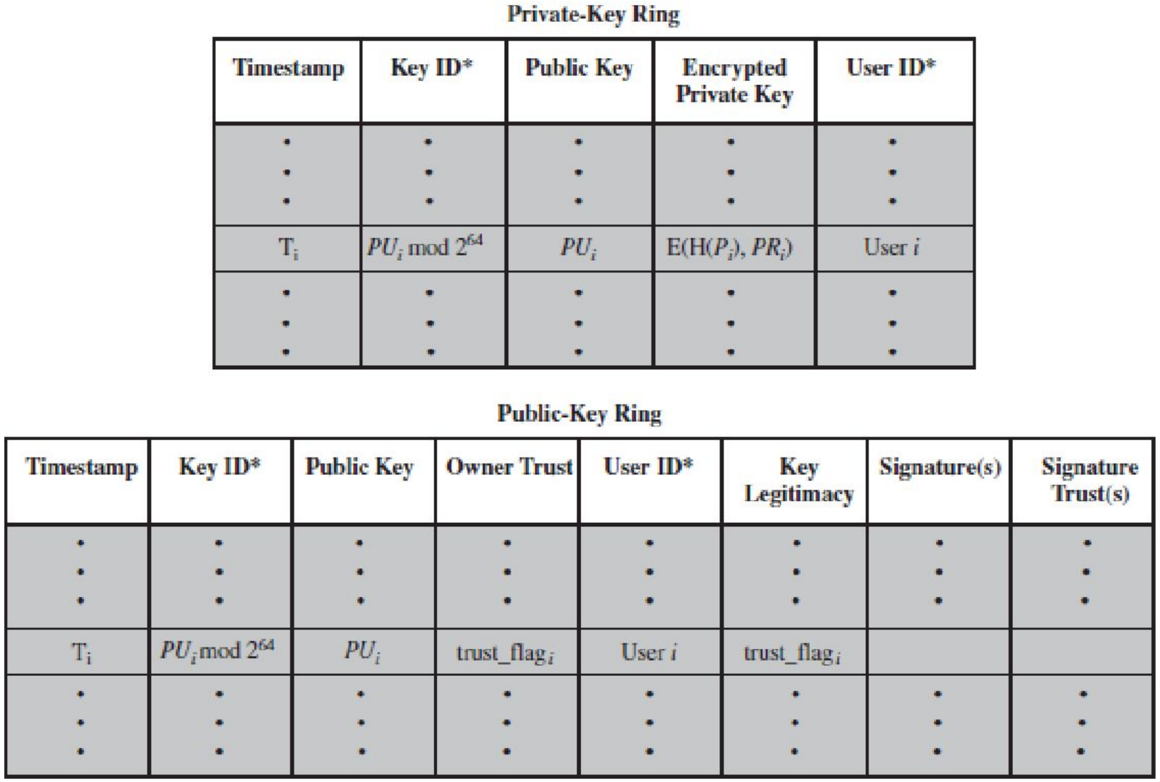
PGP Public & Private Keys

- Given that a user may have multiple public/private key pairs, how do we know which public key was used to encrypt a message.
- Send the public key along with the message.
    - Inefficient, since the key might be thousands of bits.
    - Associate a unique ID with each key pair and send that with the message. Would require that all senders know that mapping of keys to ID's for all recipients.
    - Generate an ID likely to be unique for a given user. This is PGP's solution. Use the least significant 64-bits of the key as the ID.
    - This is used by the receiver to verify that he has such a key on his "key ring." The associated private key is used for the decryption.

Key Rings

- Key IDs are critical to the operation of PGP.
- Two key IDs are included in any PGP message that provides both confidentiality and authentication.
- These keys need to be stored and organised 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, one to store the public/private key pairs owned by that node and one to store the public keys of other users known at this node.
- These data structures are referred to, respectively as the private-key ring and the public key ring.

Public /Private Key rings



Private-Key Ring

Timestamp	Key ID*	Public Key	Encrypted Private Key	User ID*
• • •	• • •	• • •	• • •	• • •
T_i	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)
• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •
T_i	PU_i mod 2^{64}	PU_i	trust_flag$_i$	User *i*	trust_flag$_i$		
• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •

Private Key Ring

- Each user maintains two key ring data structures:
  - a private-key ring for his own public/private key pairs,
  - a public-key ring for the public keys of correspondents.
  - The private key ring is a table of rows containing:
    - Timestamp: when the key pair was generated.
    - Key ID: 64 least significant digits of the public key.
    - Public key: the public portion of the key.
    - Private key: the private portion, encrypted using a passphrase.
    - User ID: usually the user’s email address. May be different for different key pairs.

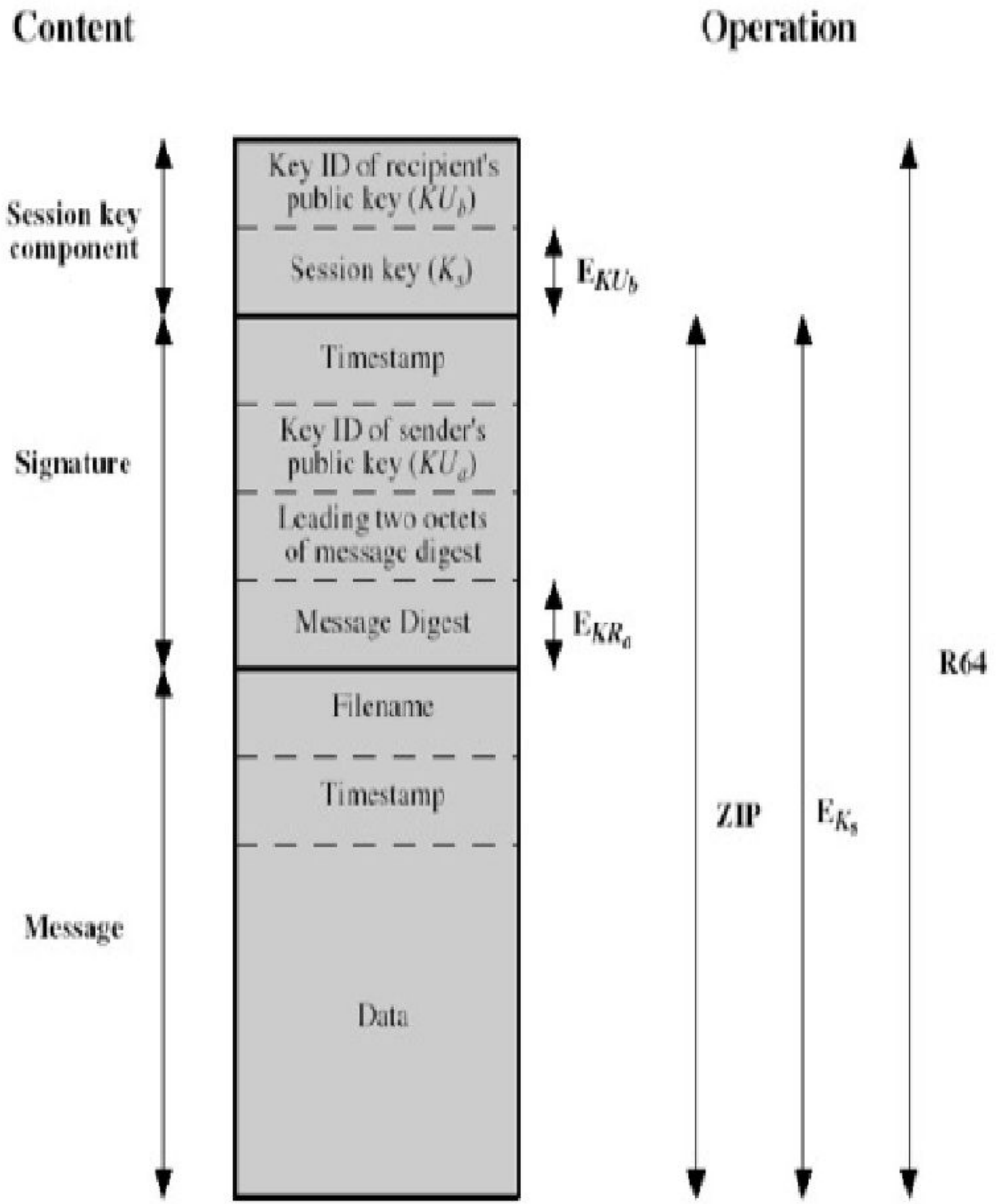
Passphrase security

- A *passphrase* is similar to a password. However, a password generally refers to something used to authenticate or log into a system. A password generally refers to a secret used to protect an encryption key. Commonly, an actual encryption key is derived from the passphrase and used to encrypt the protected resource
- PGP software is made with strong cryptography that no one, not us, not even major governments can break.
- The plus side of this is that you can always rest assured that your information is safe from prying.
- The minus side is that if you forget your password then you cannot pry into it yourself.
- The passphrase **cannot** be reset without a PGP Universal Server,
- As PGP products use a passphrase which is associated with the keypair that is used for encrypting and decrypting, it is extremely important that this passphrase is protected.
- Password stealing trojans that can monitor keystrokes exist, which make it necessary to protect the physical access to your computer.
- Use firewalls to protect your network from invasion, and use up-to-date anti-virus software to protect your passphrase from being stolen

Public Key Ring

- Public keys of other users are stored on a user's public-key ring.
- This is a table of rows containing (among other fields):
  - Timestamp: when the entry was generated.
  - Key ID: 64 least significant digits of this entry.
  - Public key: the public key for the entry.
  - User ID: Identifier for the owner of this key.

Multiple IDs may be associated with a single public key. The public key can be indexed by either User ID or Key ID.



Content

Operation

Session key component

Key ID of recipient's public key (KU_b)

Session key (K_s)

E_{KU_b}

Signature

Timestamp

Key ID of sender's public key (KU_a)

Leading two octets of message digest

Message Digest

E_{KR_a}

Message

Filename

Timestamp

Data

ZIP

E_{K_s}

R64

S/MIME

- Secure/Multipurpose Internet Mail Extension (S/MIME) is a security enhancement to the MIME Internet e-mail format standard based on technology from RSA Data Security.
- S/MIME will emerge as the industry standard for commercial and organizational use,
- while PGP will remain the choice for personal e-mail security for many users.

S/MIME

- *Secure Multipurpose Internet Mail Extensions* (S/MIME) provides a secure method of sending e-mail and is incorporated into many popular e-mail applications.
- S/MIME provides confidentiality and authentication by using the RSA asymmetric key system, digital signatures, and X.509 digital certificates.
- With S/MIME, the sender of an e-mail can provide a guarantee to the recipient that the e-mail is in fact sent from the sender, and that the content of the e-mail has not been tampered with on the way to the recipient.
- S/MIME relies on a system of public/private keys and trust authorities.
- To be able to send a signed e-mail , you need to take contact with a trust authority, such as Verisign.

S/MIME

- The trust authority establishes your identity and issues a private key and an X.509 certificate to you.
- These keys are sent to you in a password-protected file.
- The X.509 certificate may contain various information about you , like the e-mail address to which the keys are bound.
- Your public key is included in the X.509 certificate.
- When you send an e-mail to another person, you can now sign the message. When the recipient receives the signed e-mail, he/she needs to be in possession of your public key to verify the signature.
- Now that the recipient has your public key, he/she can receive signed e-mail from you and send encrypted mail to you.
- To send *encrypted* e-mail, you need the public key of the person you want to send to. You can obtain the public key of the recipient by asking him or her to send you a signed e-mail.

S/MIME (Secure/Multipurpose Internet Mail Extensions)

- Security enhancement to MIME email
  - original Internet RFC822 email was text only
  - MIME provided support for varying content types and multi-part messages
  - with encoding of binary data to textual form
  - S/MIME added security enhancements
- have S/MIME support in various modern mail agents: MS Outlook, Netscape etc
- To understand S/MIME, the recent version of this format specification is RFC 5322 (*Internet Message Format*).

S/MIME

- RFC 5322
  - A message consists of some number of header lines (*the header*) followed by unrestricted text (*the body*). The header is separated from the body by a blank line.
  - RFC 5322 defines a format for text messages that are sent using electronic mail.
  - It has been the standard for Internet-based text mail messages and remains in common use.
  - In the RFC 5322 context, 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.

S/MIME

- MIME
  - Multipurpose Internet Mail Extension (MIME) is an extension to the RFC 5322 framework that is intended to address some of the problems and limitations of the use of Simple Mail Transfer Protocol (SMTP)
  - Limitations of SMTP include
    - SMTP cannot transmit executable files or other binary files.
    - SMTP cannot transmit text data that includes national language characters because these are represented by 8-bit codes with values of 128 decimal or higher, and SMTP is limited to 7-bit ASCII.
    - SMTP servers may reject mail message over a certain size.
    - SMTP gateways that translate between ASCII to EBCDIC suffer translation problems.
    - Some SMTP implementations do not adhere completely to the SMTP standard defined in RFC 822.

MIME

MIME specification includes the following elements:

1. Five new message header fields. These fields provide information about the body of the message.
2. A number of content formats are defined, thus standardizing representations that supports multimedia e-mail.
3. Transfer encodings are defined that enable that protect any content format to be altered by the mail system.
4. MIME provides a standardized way of dealing with a wide variety of information representations in a multimedia environment.

MIME –Header

The five header fields defined in MIME are

- **MIME-Version:** Must have the parameter value 1.0. This field indicates that

the message conforms to RFCs 2045 and 2046.

- **Content-Type:** Describes the data contained in the body .
- **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:** A text description of the object with the body.

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 (contd.)

The other major component of MIME is a definition of transfer encodings for message contents:

Encoding	Description
7bit	The data are all represented by short lines of ASCII chars.
8bit	The lines are short, but there may be non-ASCII chars.
Binary	Not only may non-ASCII chars be presented but 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 to 8-bit printable ASCII characters blocks.
x-token	A nonstandard encoding.

S/MIME Functions

- **Enveloped data:** This consists of encrypted content of any type and encrypted content encryption keys for one or more recipients.
- **Signed data:** A digital signature is formed by taking the message digest of the content to be signed and then encrypting that with the private key of the signer. The content plus signature are then encoded using base64 encoding. A signed data message can only be viewed by a recipient with S/MIME capability.
- **Clear-signed data:** As with signed data, a digital signature of the content is formed. However, in this case, only the digital signature is encoded using base64.As a result, recipients without S/MIME capability can view the message content, although they cannot verify the signature.
- **Signed and enveloped data:** Signed-only and encrypted-only entities may be nested, so that encrypted data may be signed and signed data or clear-signed data may be encrypted.

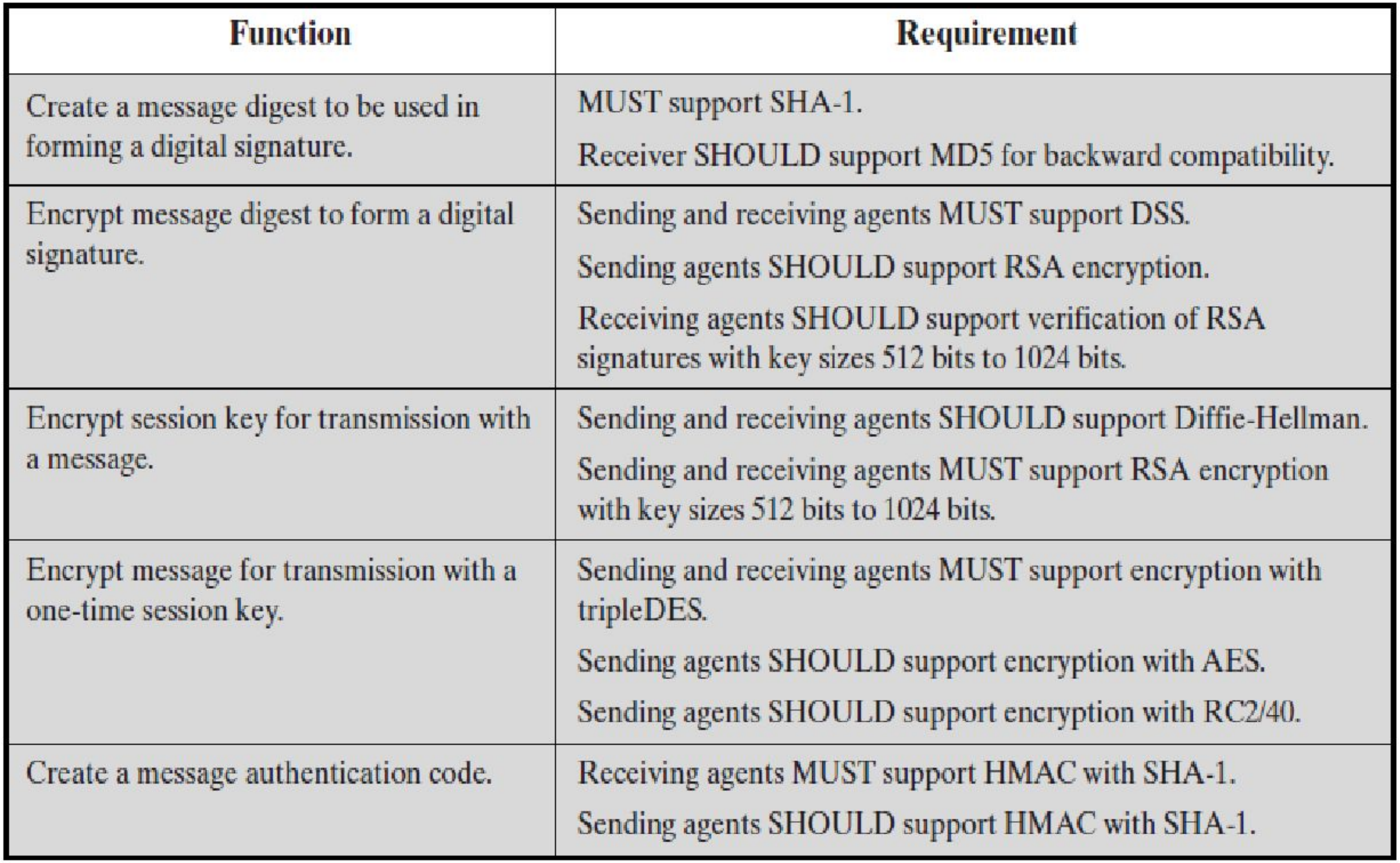
S/MIME Cryptography

Definitions:

MUST: The definition is an absolute requirement of the specification.

SHOULD: There may exist valid reasons in particular circumstances to ignore this feature or function, but it is recommended that an implementation include the feature or function.

Cryptographic algorithm 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.