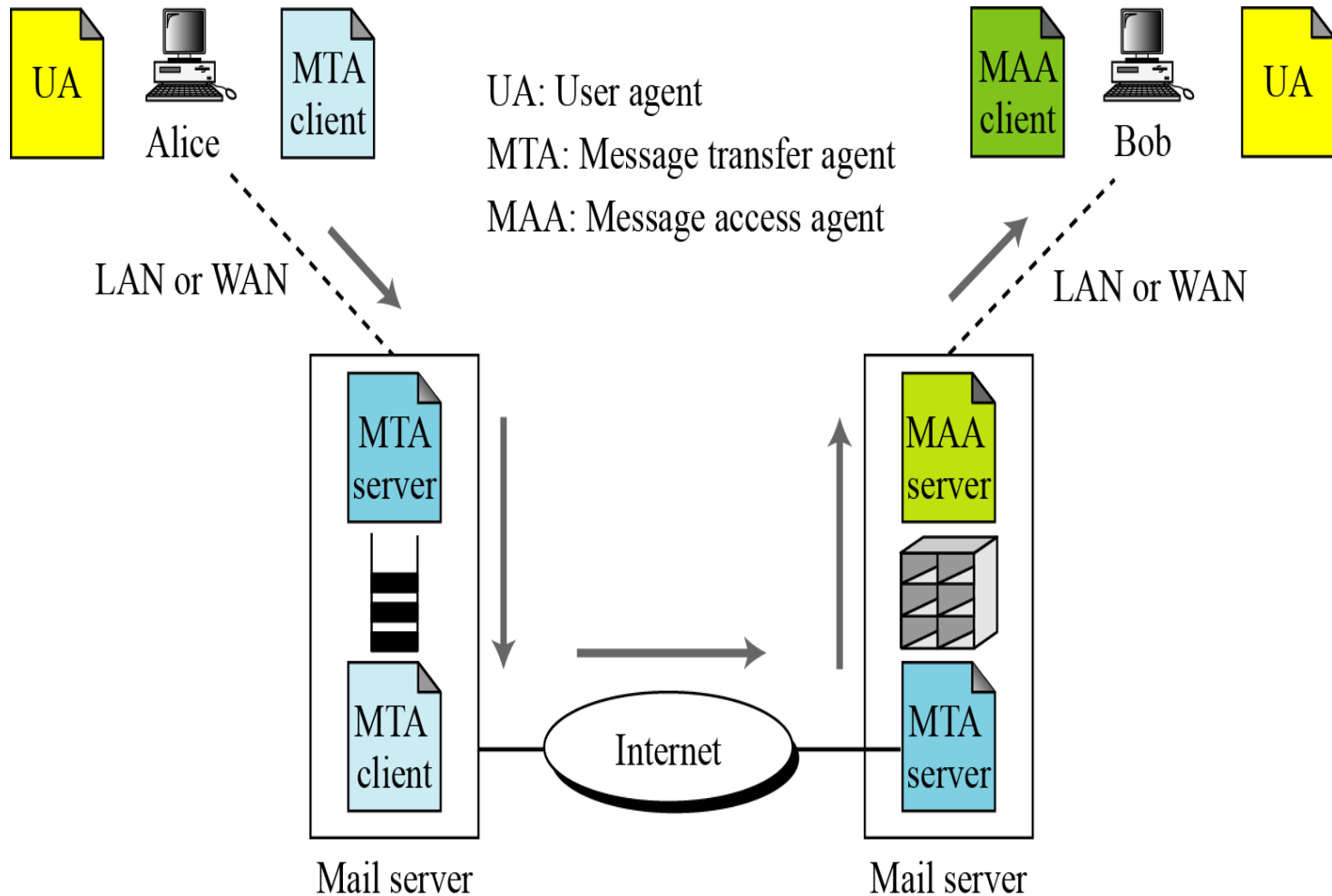


# E-mail Security(S/MIME)

# *E-mail Architecture*



# E-mail Security

- E-mail is one of the most widely used network services
  - killer application of the Internet
- Normally message contents not secured
  - Can be read/modified either in transit or at destination by the attacker
- E-mail service is like postcard service
  - just pick it and read it

# 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

# S/MIME

- Secure/Multipurpose Internet Mail Extensions
- A standard way for email encryption and signing
- IETF effort (RFCs 2632, 2633 – for version 3.0; RFCs 3850, 3851 for version 3.1; 5750, 5751 for version 3.2)
- Industry support
- Not a standalone software, a system that is to be supported by email clients
  - such as MS Outlook and Thunderbird
- S/MIME handles digital signatures
  - Also provides encryption

# Quick E-mail History

- SMTP and RFC 822
  - only ASCII messages (7-bit)
- MIME (Multipurpose Internet Mail Extensions)
  - content type
    - Almost any type of information can appear in an email message
  - transfer encoding
    - specifies how the message body is encoded into textual form (radix64 is common)
- S/MIME: Secure MIME
  - new content types, like signature, encrypted data

# S/MIME Cryptographic Algorithms

- hash functions: SHA-1 & MD5
- digital signatures: DSS & RSA
- session key encryption: ElGamal & RSA
- message encryption: Triple-DES, AES and others
- sender should know the capabilities of the receiving entity (public announcement or previously received messages from receiver)
  - otherwise sender takes a risk

# Scope of S/MIME Security

- S/MIME secures a MIME entity
  - a MIME entity is entire message except the headers
  - so the header is not secured
- First MIME message is prepared
- This message and other security related data (algorithm identifiers, certificates, etc.) are processed by S/MIME
- and packed as one of the S/MIME content type





## E-mail Format

E-mail header
MIME-Version: 1.1 Content-Type: type/subtype Content-Transfer-Encoding: encoding type Content-Id: message id Content-Description: textual explanation of nontextual contents
E-mail body

MIME headers

# MIME - New header fields

- **MIME-Version**
- **Content-Type**
  - describes the data contained in the body
  - receiving agent can pick an appropriate method to represent the content
- **Content-Transfer-Encoding**
  - indicates the type of the transformation that has been used to represent the body of the message
- **Content-ID**
- **Content-Description**
  - description of the object in the body of the message
  - useful when content is not readable (e.g., audio data)



## *MIME-Version*

*This header defines the version of MIME used. The current version is 1.1.*

**MIME-Version: 1.1**

## *Content-Type*

*The content type and the content subtype are separated by a slash. Depending on the subtype, the header may contain other parameters.*

**Content-Type: <type / subtype; parameters>**

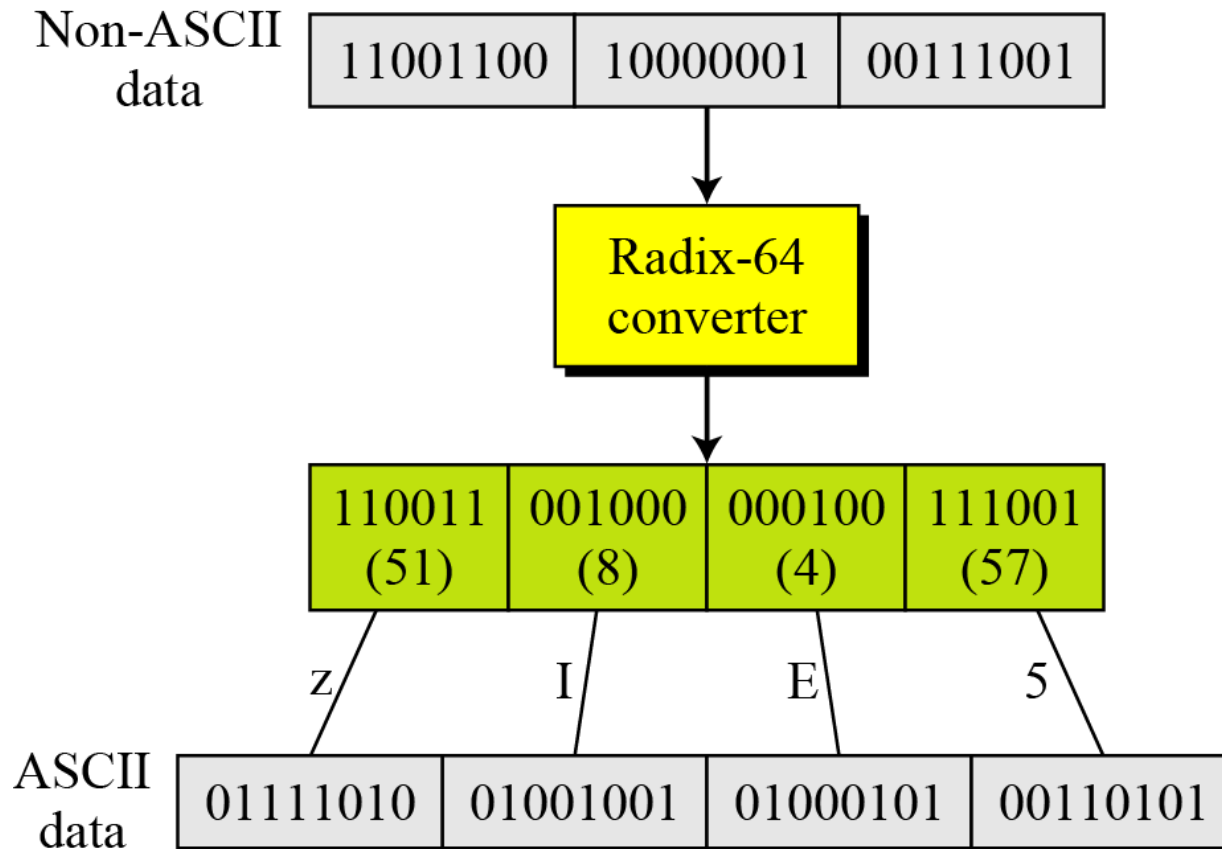
**Table 16.14** *Data types and subtypes in MIME*

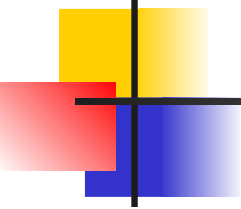
<i>Type</i>	<i>Subtype</i>	<i>Description</i>
Multipart	Plain	Unformatted.
	HTML	HTML format.
	Mixed	Body contains ordered parts of different data types.
	Parallel	Same as above, but no order.
	Digest	Similar to Mixed, but the default is message/RFC822.
Message	Alternative	Parts are different versions of the same message.
	RFC822	Body is an encapsulated message.
	Partial	Body is a fragment of a bigger message.
	External-Body	Body is a reference to another message.
Image	JPEG	Image is in JPEG format.
	GIF	Image is in GIF format.
Video	MPEG	Video is in MPEG format.
Audio	Basic	Single channel encoding of voice at 8 KHz.
Application	PostScript	Adobe PostScript.
	Octet-stream	General binary data (eight-bit bytes).

# MIME – Transfer encodings

- **7bit**
  - short lines of ASCII characters
- **8bit**
  - short lines of non-ASCII characters
- **binary**
  - non-ASCII characters
  - lines are not necessarily short
- **quoted-printable**
  - non-ASCII characters are converted into hexa numbers
- **base64 (radix 64)**
  - 3 8-bit blocks into 4 6-bit blocks

# *Radix-64 conversion*



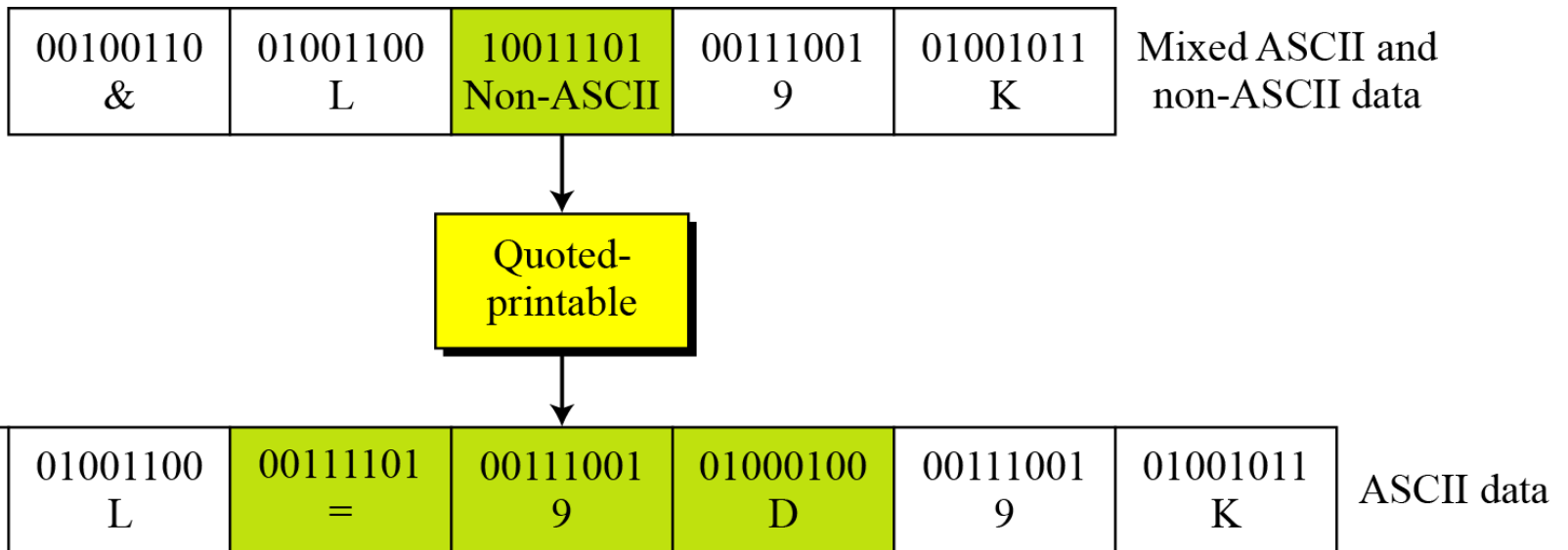


**Table 16.16** *Radix-64 encoding table*

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



## *Quoted-printable*








# S/MIME Functions

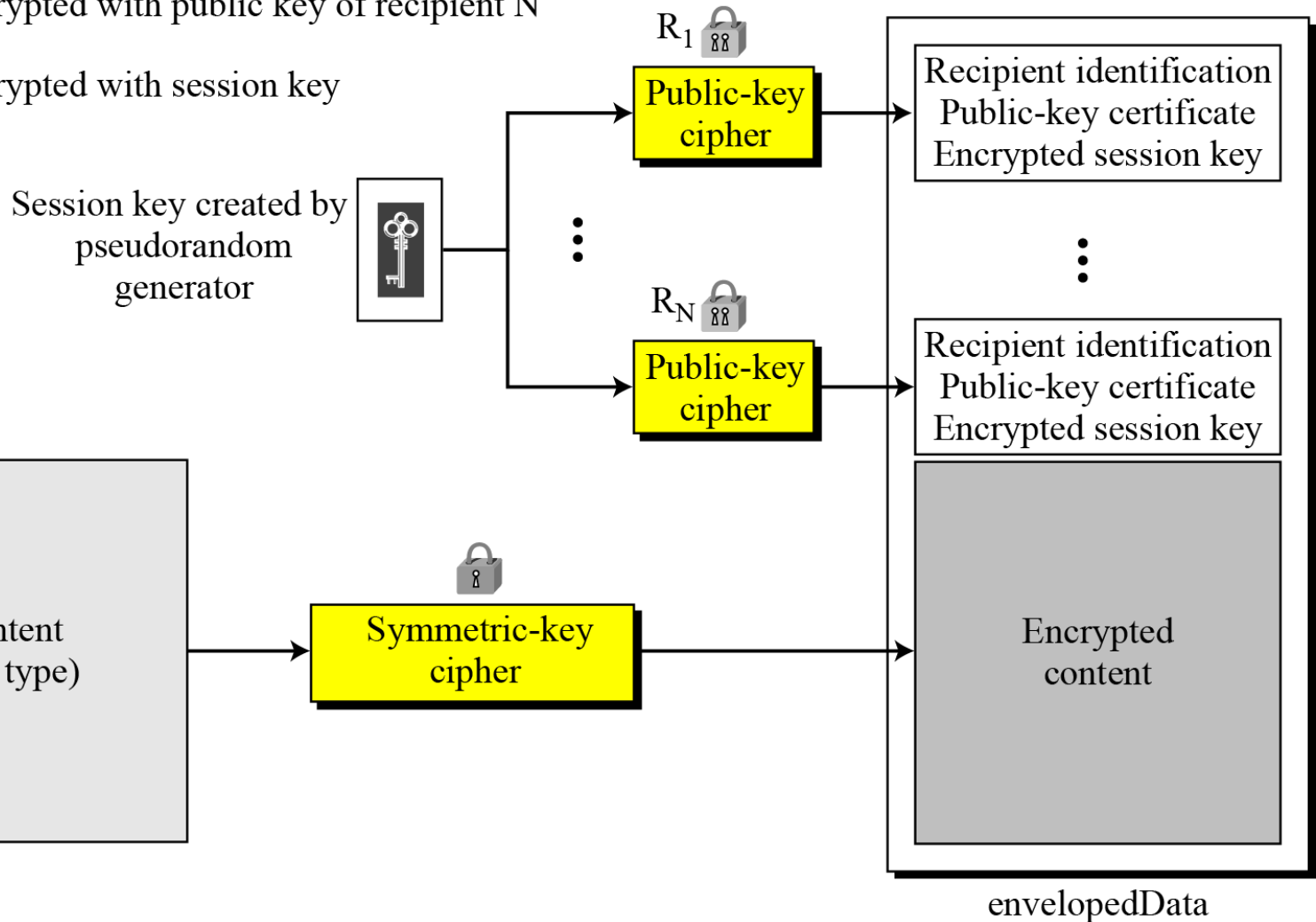
- enveloped data
  - encrypted content and associated keys
- signed data
  - encoded message + encoded signed message digest
- clear-signed data
  - cleartext message + encoded signed message digest
- authenticated data
  - encoded message + encoded signed message digest which is encrypted

# *Enveloped-data content type*

$R_1$   Encrypted with public key of recipient 1

$R_N$   Encrypted with public key of recipient N

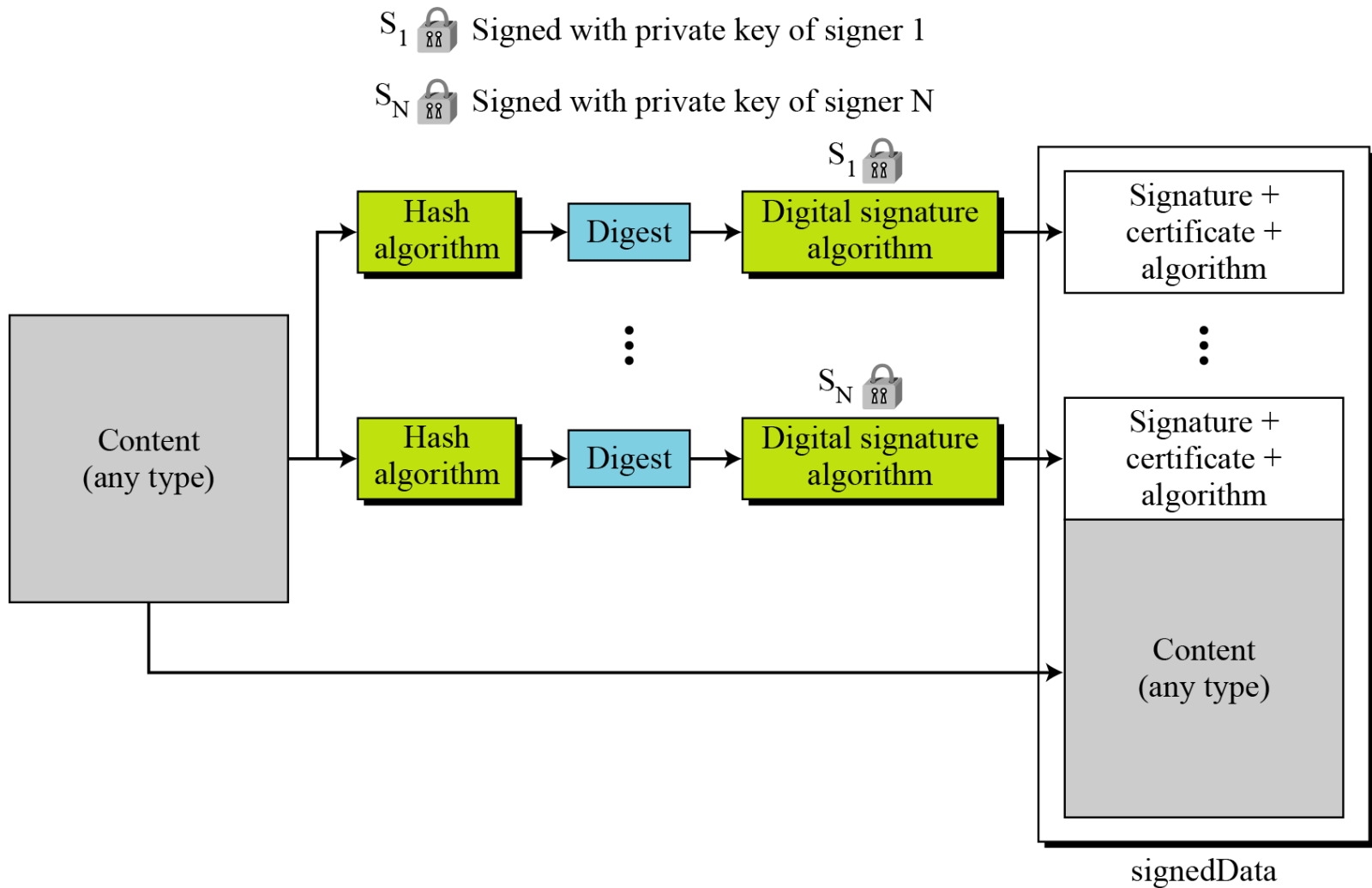
 Encrypted with session key



# EnvelopedData

- For message encryption
- Similar to PGP
  - create a random session key, encrypt the message with that key and a conventional crypto, encrypt the session key with recipient's public key
- Unlike PGP, recipient's public key comes from an X.509 certificate
  - trust management is different

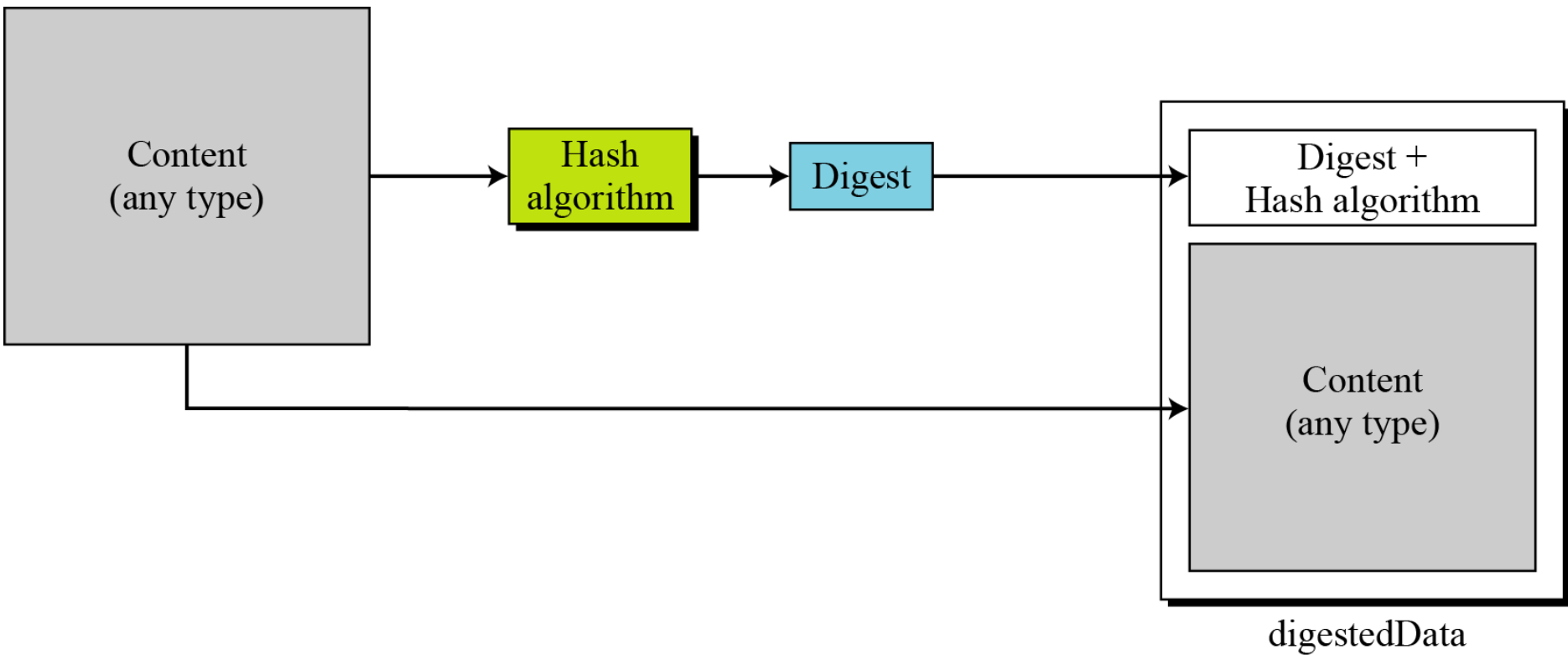
## *Signed-data content type*



# SignedData

- For signed message
  - both message and signature are encoded so that the recipient only sees some ASCII characters if he does not use an email client with S/MIME support
- Similar to PGP
  - first message is hashed, then the hash is encrypted using sender's private key
- Message, signature, identifiers of algorithms and the sender's certificate are packed together
  - again difference between S/MIME and PGP in trust management


## clear-signed data *content type*




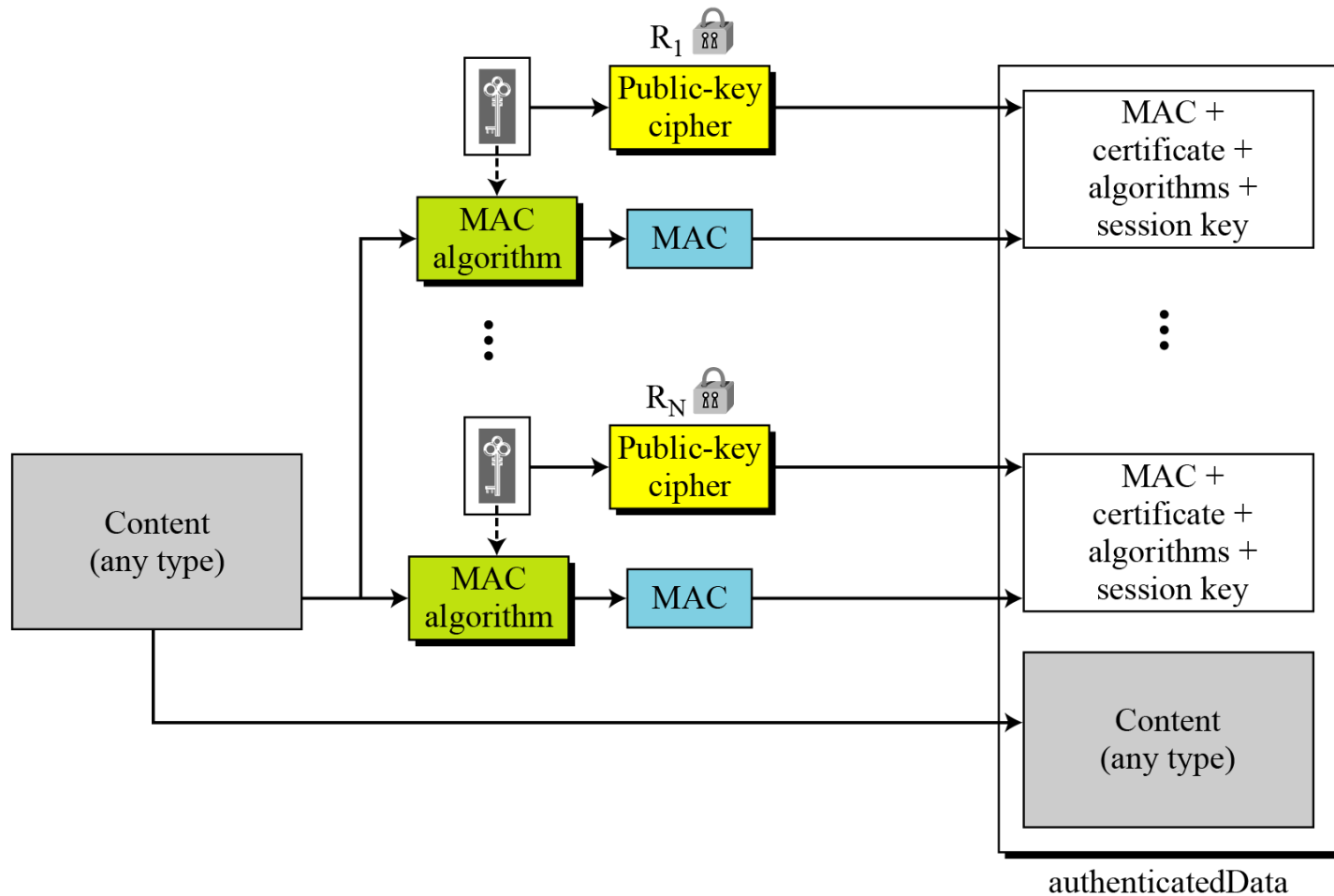
# Clear Signing

- Another mechanism for signature
  - but the message is not encoded, so an email client with no S/MIME support could also view the message
    - of course the signature will not be verified and will be seen as a meaningless attachment
- multipart/signed content type
  - 2 parts
    - Clear text message
    - Signature

# Authenticated-data content type

$R_1$   Encrypted with public key of recipient 1

$R_N$   Encrypted with public key of recipient N





# Cryptographic Algorithms

*S/MIME defines several cryptographic algorithms.*

*The term “must” means an absolute requirement;  
the term “should” means recommendation.*

**Table 16.17** *Cryptographic algorithm for S/MIME*

<i>Algorithm</i>	<i>Sender must support</i>	<i>Receiver must support</i>	<i>Sender should support</i>	<i>Receiver should support</i>
Content-encryption algorithm	Triple DES	Triple DES		1. AES 2. RC2/40
Session-key encryption algorithm	RSA	RSA	Diffie-Hellman	Diffie-Hellman
Hash algorithm	SHA-1	SHA-1		MD5
Digest-encryption algorithm	DSS	DSS	RSA	RSA
Message-authentication algorithm		HMAC with SHA-1		



## Example 16.3

# Enveloped data – Example

**The following shows an example of an enveloped-data in which a small message is encrypted using triple DES.**

**Content-Type:** application/pkcs7-mime; mime-type=enveloped-data

**Content-Transfer-Encoding:** Radix-64

**Content-Description:** attachment

**name=“report.txt”;**

cb32ut67f4bhijHU21oi87eryb0287hmnklsgFDoY8bc659GhIGfH6543mhjkdsaH23YjBnmN  
ybmlkzjhgfdyhGe23Kjk34XiuD678Es16se09jy76jHuytTMDcbnmlkjgfFdiuyu678543m0n3h  
G34un12P2454Hoi87e2ryb0H2MjN6KuyrlsgFDoY897fk923jljk1301XiuD6gh78EsUyT23y

# Clear-signed data – Example

Content-Type: multipart/signed; protocol="application/pkcs7-signature";  
micalg=sha1; boundary=boundary42

--boundary42

Content-Type: text/plain

This is a clear-signed message.

--boundary42

Content-Type: application/pkcs7-signature; name=smime.p7s

Content-Transfer-Encoding: base64

Content-Disposition: attachment; filename=smime.p7s

ghyHhHUujhJhjH77n8HHGTrfvbnj756tbB9HG4VQpfyF467GhIGfHfYT6  
4VQpfyF467GhIGfHfYT6jH77n8HHGghyHhHUujhJh756tbB9HGTrfvbnj  
n8HHGTrfvhJhjH776tbB9HG4VQbnj7567GhIGfHfYT6ghyHhHUujpfyF4  
7GhIGfHfYT64VQbnj756

--boundary42--

# S/MIME Certificate Processing

- S/MIME uses X.509 v3 certificates
  - Certification Authorities (CAs) issue certificates
  - unlike PGP, a user cannot be a CA
- each client has a list of trusted CA certificates
  - actually that list comes with e-mail client software or OS and own public/private key pairs and certs